

UNIVERSITAT POLITÈCNICA DE CATALUNYA
DEPARTAMENT DE LLENGUATGES I SISTEMES INFORMÀTICS
MÀSTER EN COMPUTACIÓ

TESI DE MÀSTER

LES RESTRICCIONS D'INTEGRITAT TEMPORALS CONSTANT I PERMANENT

ESTUDIANT: RAQUEL PAU FERNÁNDEZ
DIRECTOR(S): ANTONI OLIVÉ RAMON
PONENT: ANTONI OLIVÉ RAMON

DATA: 25/06/2007

Index

PRELIMINARS	5
1. INTRODUCCIÓ	6
1.1 RESUM	6
1.2 MOTIVACIÓ	6
1.3 OBJECTIUS I METODOLOGIA	7
1.4 ESTRUCTURA DE LA TESI	7
2. ESTAT DE L'ART DE LES RESTRICCIONS D'INTEGRITAT TEMPORALS	9
2.1 INTRODUCCIÓ	9
2.2 CONCEPTES I NOTACIÓ PRÈVIA	9
2.3 RESTRICCIONS D'INTEGRITAT	12
2.3.1 DEFINICIÓ	12
2.3.2 CAUSES DE VIOLACIÓ	13
2.3.3 TIPUS DE CLASSIFICACIÓ	14
2.4 CLASSIFICACIÓ DE RESTRICCIONS D'INTEGRITAT DE TRANSICIÓ	15
2.4.1 RESTRICCIONS DE TRANSICIÓ UNIVERSAL	15
2.4.2 RESTRICCIONS DE TRANSICIÓ EXISTENCIAL	16
2.4.3 FREQUÈNCIA DE TIPUS D'ENTITAT I DE RELACIÓ	16
2.4.4 TIPUS D'ENTITAT I DE RELACIÓ PERMANENTS	16
2.4.5 TIPUS D'ENTITAT I DE RELACIÓ CONSTANT	16
2.4.6 PARTICIPACIÓ INICIAL D'UN PARTICIPANT EN UN TIPUS DE RELACIÓ	17
2.4.7 INTERVAL D'EXISTÈNCIA D'UN TIPUS DE RELACIÓ EN UN PARTICIPANT	17
2.4.8 MUTABILITAT D'UN TIPUS DE RELACIÓ RESPECTE UN PARTICIPANT	18
2.4.9 INSERIBILITAT EN UN TIPUS DE RELACIÓ RESPECTE D'UN PARTICIPANT	18
2.5 LES RESTRICCIONS D'INTEGRITAT TEMPORALS EN LES TAXONOMIES	19
2.1.1 ESPECIALITZACIÓ I GENERALITZACIÓ	19
2.1.2 COVERING I DISJOINTNESS	20
2.1.3 RESTRICCIONS D'EVOLUCIÓ D'ESPECIALITZACIONS	20
2.1.4 VALID TYPE CONFIGURATION TRANSITIONS	22
2.2 RESTRICCIONS D'INTEGRITAT TEMPORAL AMB Lògica Temporal	23
2.2.1 INTRODUCCIÓ	23
2.2.2 HISTÒRIA DE LA Lògica Temporal	24
2.2.3 Lògica Temporal: Modelat del Temps	29
2.2.4 CLASSIFICACIONS DE LES Lògiques Temporals	30
2.3 NOTES BIBLIOGRÀFIQUES	33
3. RESTRICCIONS D'INTEGRITAT CONSTANT I PERMANENT	34
3.1 INTRODUCCIÓ	34
3.2 DEFINICIÓ DE LES RESTRICCIONS DINÀMIQUES CONSTANT I PERMANENT	34
3.2.1 CONSIDERACIONS PRÈVIES.	34
3.2.2 APLICADES A UN TIPUS D'ENTITAT	35
3.2.3 APLICADES A UN PARTICIPANT D'UN TIPUS DE RELACIÓ	36
3.2.4 APLICADES A UN TIPUS DE RELACIÓ	38

3.3 LES RESTRICCIONS TEMPORALS CONSTANT I PERMANENT APLICADES A LES REIFICACIONS	39
3.3.1 INTRODUCCIÓ	39
3.3.2 SEMÀNTICA DE LES RESTRICCIONS CONSTANT I PERMANENT APLICADES A UNA REIFICACIÓ	40
3.4 PROPIETATS RELATIVES A L'ESQUEMA ESTRUCTURAL DE LES RESTRICCIONS TEMPORALS CONSTANT I PERMANENT	41
3.4.1 RELATIVES ALS TIPUS DE RELACIONS	41
3.4.2 RELATIVES A REIFICACIONS	45
3.5 PROPIETATS RELATIVES A L'ESQUEMA DE COMPORTAMENT DE LES RESTRICCIONS TEMPORALS CONSTANT I PERMANENT	51
3.5.1 CONCEPTES PREVIS	51
3.5.2 PROPIETATS	52
3.6 NOTES BIBLIOGRÀFIQUES	53
 <u>4. IMPORTÀNCIA PRÀCTICA DE LES RESTRICCIONS CONSTANT I PERMANENT.</u>	 54
4.1 INTRODUCCIÓ	54
4.2 CASOS D'ESTUDI	55
4.2.1 REENGINYERIA D'UN SISTEMA DE MERCAT ELECTRÒNIC: EBAY	55
4.2.2 EU-RENT CAR RENTALS SPECIFICATION	56
4.2.3 DBLP	58
4.3 NOTACIÓ I RESULTATS	58
4.4 CONCLUSIONS	59
4.5 NOTES BIBLIOGRÀFIQUES	59
 <u>5 REPRESENTACIÓ PROPOSADA EN UML DE LES RESTRICCIONS CONSTANT I PERMANENT</u>	 60
5.1 INTRODUCCIÓ	60
5.2 INTRODUCCIÓ AL METAESQUEMA D' UML	61
5.3 REPRESENTACIÓ DE LES RESTRICCIONS TEMPORALS CONSTANT I PERMANENT EN UML 1.4	62
5.3.1 CONCEPTES PREVIS	62
5.3.2 FRAGMENTS DEL METAESQUEMA	63
5.3.3 LA CANVIABILITAT	64
5.3.4 NOTACIÓ	64
5.3.5 EQUIVALÈNCIA AMB LES RESTRICCIONS TEMPORALS CONSTANT I PERMANENT	65
5.4 REPRESENTACIÓ DE LES RESTRICCIONS TEMPORALS CONSTANT I PERMANENT EN UML 2.1.1	66
5.4.1 CONCEPTES PREVIS	66
5.4.2 FRAGMENTS DEL METAESQUEMA	66
5.4.3 LA CANVIABILITAT	67
5.4.4 NOTACIÓ	67
5.4.5 EQUIVALÈNCIA AMB LES RESTRICCIONS TEMPORALS CONSTANT I PERMANENT	68
5.5 REPRESENTACIÓ PROPOSADA DE LES RESTRICCIONS TEMPORALS CONSTANT I PERMANENT EN UML	68
5.5.1 REPRESENTACIÓ D'ENTITATS AMB POBLACIÓ CONSTANT O PERMANENT	68
5.5.2 REPRESENTACIÓ DE PARTICIPACIONS CONSTANTS O PERMANENTS.	69
5.6 NOTES BIBLIOGRÀFIQUES	70

6	<u>CONCLUSIONS</u>	<u>71</u>
7	<u>BIBLIOGRAFIA</u>	<u>72</u>
	<u>ANNEXOS</u>	<u>74</u>
7.1	CAS D'ESTUDI EU-RENTCAR	75
7.2	CAS D'ESTUDI EBAY	93
7.3	CAS D'ESTUDI DBLP	108
7.4	ARTICLE PRESENTAT I ACCEPTAT EN JISBD07	113

Preliminars

Agraïments

Voldria donar gràcies a l'entusiasme pel treball i l'ajuda del tutor Antoni Olivé i als comentaris de Cristina Gómez. També voldria donar gràcies a la confiança i paciència de la meva família i sobretot a dos companys que malgrat tot m'han ajudat a arribar fins aquí: Alberto Coroleu i Antonio Quesada. També dono gràcies pels moments que m'han donat Jorge Luis Rodríguez, Marta Lozano, Daniel Cabrera, Cecilia González, Carlos Herrera, Sergi Gòmez, Ivan Mingueza, Daniel Albert, Jordi Manrique i Isaac Jurado.

1. Introducció

1.1 Resum

Aquesta tesi de màster tracta de l'estudi de les restriccions temporals constant i permanent. Aquestes restriccions s'apliquen a tipus d'entitat, a tipus de relacions i a participacions en tipus de relació. En aquest treball es proporciona una definició formal d'aquestes restriccions empleant lògica de primer ordre i lògica temporal. A continuació s'analitzen les seves propietats i es realitza l'estudi de tres casos reals amb l'objectiu de conèixer la seva importància pràctica. La conclusió a la que s'arriba és que apareixen molt sovint, i per tant justifica l'estudi i l'ús d'una notació específica en UML que faciliti la seva definició. Els metamodels de les versions 1.4 i 2.1 del nucli d'UML no permeten indicar explícitament les restriccions constant i permanent, i per tant, es modelaran mitjançant estereotips. Finalment es dona uns possibles patrons de disseny per garantir el compliment d'aquestes restriccions en temps d'execució.

1.2 Motivació

D'acord amb el principi del 100%, o de completesa[16], un esquema conceptual ha d'incloure la definició de totes les restriccions d'integritat (per abreviar, restriccions) que han de satisfer la base d'informació. Per aquest motiu, tots els llenguatges complets de modelització conceptual inclouen un (sub)llenguatge per definir restriccions. També, i consegüentment amb el principi del 100%, un metamodel ha d'especificar el conjunt de propietats i restriccions dels conceptes que permet representar i així poder-se validar una instància d'aquest, es a dir, un esquema conceptual.

En el cas general, una restricció pot ser qualsevol expressió vàlida del llenguatge corresponent. No obstant, hi ha algunes classes particulars de restriccions que es defineixen en molts esquemes conceptuais i molt sovint en un mateix esquema. Naturalment, aquestes restriccions poden definir-se en el llenguatge general de definició de restriccions, però resulta pràctic utilitzar construccions lingüístiques especials per a elles que facilitin la seva definició.

Les classes més importants són les restriccions d'integritat referencial i les de cardinalitat, que ja estaven identificades en l'article original del model entitat-relació[5] i que tots els llenguatges tracten de forma especial. El llenguatge que probablement identifica més restriccions particulars és l'ORM[9].

Les restriccions particulars més populars són estàtiques. No obstant, hi ha algunes restriccions temporals que apareixen tant sovint en els esquemes conceptuals que també justifiquen un tractament especial. En aquesta tesi estudiem dos, que anomenem constant i permanent, i que poden aplicar-se a tipus d'entitat, a tipus de relació i a participacions en tipus de relació.

Aquestes restriccions no són noves. Es mencionen amb alguna variació en diversos texts, però no existeix una definició completa i formal.

1.3 Objectius i metodologia

El primer objectiu d'aquesta tesi és donar una **definició completa i formal** de les restriccions constant i permanent, tant en lògica temporal com en lògica de primer ordre. Un cop la definició quedi formulada, es realitzarà un estudi de les propietats d'aquestes restriccions. Aquest consistirà en examinar quines conseqüències apareixen en els diferents elements estructurals i de comportament. En paral·lel realitzarem un estudi sobre la seva importància pràctica, el qual consistirà en fer un recompte del nombre d'aparicions d'aquestes restriccions en diferents esquemes conceptuals i per tant, poder concloure si seria convenient disposar d'una notació gràfica per representar-les.

Cal remarcar, que fins ara els esquemes conceptuals amb UML no feien explícites les restriccions constant i permanent. Per tant, aquestes quedaven relativament garantides pels events de domini definits en l'esquema de comportament. Això pot provocar que en futures modificacions d'aquests treballs aquestes restriccions puguin ser violades pel seu desconeixement.

A continuació, i com últim objectiu d'aquesta tesi, sigui quina sigui la freqüència d'aparició d'aquestes restriccions, atès que l'OCL no disposa d'una semàntica temporal reconeguda i el principi del 100%, s'estudiarà si en les versions 1.4 i 2.1 d'UML es disposa de **notació** específica, per definir aquestes restriccions i quina seria una representació vàlida en la versió 2.1.

1.4 Estructura de la tesi

Atès que en aquest capítol s'ha definit quin és el context del treball i quins són els objectius a complir, la tesi queda estructurada pels següents capítols:

- **Estat de l'art de les restriccions d'integritat temporals:** Disposa de dos seccions. Per una banda estudia les restriccions temporals en modelització conceptual i per l'altra explica com representar mitjançant lògica temporal o lògica de primer ordre fets amb semàntica temporal.
- **Restriccions d'integritat temporal constant i permanent:** Inclou la seva definició formal i el recull de les seves propietats amb la corresponent demostració.
- **Importància pràctica de les restriccions constant i permanent:** Recull els resultats obtinguts i les conclusions dels diferents casos d'estudi. La justificació i decisió sobre si un tipus d'entitat, relació o participació en un esquema estructural és constant o permanent es pot trobar en els annexos.

- **Representació en UML proposada de les restriccions constant i permanent:**
Explora els elements del llenguatge de les versions 1.4 i 2.1 d'UML relacionats amb les restriccions d'integritat constant i permanent i explora una possible representació.
- **Conclusions**
- **Annexos**

2. Estat de l'art de les restriccions d'integritat temporals

2.1 Introducció

Aquest capítol és l'estat de l'art de les restriccions d'integritat temporal, o també anomenades dinàmiques o de transició. Aquestes restriccions no són noves. Es mencionen amb alguna variació en diversos textos (com, per exemple, en [7]) i són recollides en [10], però no existeix una definició completa i formal. Aquest capítol correspondrà a la teoria necessària per entendre la definició ja existent de les restriccions constant i permanent i els conceptes bàsics de lògica temporal que ens ajudaran a expressar formalment aquestes restriccions en el capítol 3.

L'estructura serà la següent: en el punt 2.2 es defineixen els conceptes bàsics que tenen una noció temporal en modelització conceptual. A continuació, en el punt 2.3 es defineixen les restriccions d'integritat, donant pas al punt 2.4 que estudia un tipus de restriccions d'integritat: les de transició. En el punt 2.5 s'expliquen les restriccions que afecten a les taxonomies i a la seva evolució. Per finalitzar en el punt 2.6 s'expliquen la lògica temporal com a llenguatge per especificar restriccions temporals.

2.2 Conceptes i notació prèvia

El temps en modelització conceptual obliga a fer referència a diferents estats d'una base d'informació, per exemple per definir restriccions d'integritat temporals. Per aquest motiu, ha calgut definir un conjunt de conceptes a nivell de modelització conceptual. A continuació mostrem aquells conceptes que ens poden ajudar en la definició de les restriccions dinàmiques constant i permanent.

Vida del sistema $LS(t_i, t_f)$

Les entitats i relacions són instàncies d'un tipus en punts de temps determinats, els quals són expressats en una unitat de temps base com podrien ser els segons o els dies. Llavors, definim



Figura 1 Exemple d'interval de classificació

que la vida - *life span* - d'un sistema d'informació com un interval temporal $Ls(t_i, t_f)$ en el que el sistema d'informació existeix.

Instàncies $E(e, t)$ i $R(e_1, \dots, e_n, t)$

La representació de $E(e, t)$ correspon fet que una entitat e sigui instància del tipus d'entitat E en un instant de temps t (o estat).

Donada un tipus relació $R(p_1:E_1, p_2:E_2, \dots)$, on p_1, p_2, \dots, p_n són els rols, i E_1, E_2, \dots , En són els tipus d'entitats que hi participen, la representació de $R(e_1, e_2, \dots, e_n, t)$ correspon fet que hi hagi una relació entre les entitats e_1, e_2, \dots, e_n el temps t , del tipus E_1, E_2, \dots, E_n . Els atributs són considerats tipus de relacions binàries. Si s'omet un rol p_i es suposa que és igual al tipus d'entitat del participant E_i començant per minúscula.

Restricció d'integritat temporal

Sembla evident, però per a que es satisfaci el fet que $E(e, t)$, t ha de ser inclòs en la vida del sistema. Aquesta condició anomenada restricció d'integritat temporal ha de ser satisfeta per cada tipus d'entitat i formalment és representada:

$$E(e, t) \rightarrow \text{Inclòs}(t, Ls)$$

De forma equivalent succeiria amb els fets relatius a les relacions:

$$R(e_1, \dots, e_n, t) \rightarrow \text{Inclòs}(t, Ls)$$

En aquest cas és implícit la necessitat d'una restricció de que e_i , per a $i=1..n$, siguin instància de E_i en el temps t o anterior a ell. Aquesta última restricció rep el nom de restricció d'integritat referencial temporal.

Interval de classificació $Ci(e, E)$

En el cas d'un tipus d'entitat E , un interval de classificació és el conjunt màxim d'instants consecutius durant el qual la instància e és instància de E . El conjunt $Ci(e, E)$ denota tots els intervals de classificació de e en E .

De forma equivalent, un interval de classificació de (e_1, \dots, e_n) en R és el conjunt màxim d'instants consecutius durant el qual (e_1, \dots, e_n) és instància de R .

En el cas d'una relació, designem per $Ci((e_1, \dots, e_n), R)$ el conjunt d'interval·ls de classificació de (e_1, \dots, e_n) en R , i $Ci(e_i, E_i)$ el conjunt d'interval·ls de classificació de e_i a E_i .

Restricció d'integritat referencial temporal

Cada tipus de relació $R(p_1:E_1, \dots, p_n:E_n)$ en l'esquema conceptual ha d'incloure, en general, n restriccions d'integritat referencial temporal.

$$R(e_1, \dots, e_i, \dots, e_n, t) \rightarrow t_1 (E_1(e_1, t_1) \wedge t_1 \leq t)$$

....

$$R(e_1, \dots, e_i, \dots, e_n, t) \rightarrow t_n (E_n(e_n, t_n) \wedge t_n \leq t)$$

que garanteixen que les entitats participants en una relació siguin efectivament instàncies del seu tipus. Observis que en les restriccions d'integritat referencial temporal no s'exigeix que les $e_1, \dots, e_i, \dots, e_n$ siguin instància de $E_1, \dots, E_i, \dots, E_n$ en t , sinó simplement que hagin estat en algun instant fins t .

Si un participant d'un tipus de relació és un tipus d'entitat constant, llavors la corresponent restricció d'integritat referencial es pot ometre, perquè és necessàriament certa. Per exemple, en el tipus de relació:

Estoc(Article, Botiga, Natural, Temps),

la qual representa les unitats d'estoc que hi ha en una botiga d'un cert article a en un instant de temps, les restriccions d'integritat referencial temporal són:

$$\text{Estoc}(a, b, 4, t) \rightarrow t_1 | \text{Article}(a, t_1) \wedge t_1 \leq t$$

$$\text{Estoc}(a, b, 4, t) \rightarrow t_1 | \text{Botiga}(b, t_1) \wedge t_1 \leq t$$

No hi ha la corresponent a *Natural* perquè és un tipus de dades i, per tant, té població constant.

Sincronisme

Sigui el tipus de relació $R(p_1:E_1, \dots, p_n:E_n)$. La restricció d'integritat referencial temporal aplicada a R , exigeix que, si $R(e_1, \dots, e_n, t)$ és cert, llavors e_1, \dots, e_n han de ser instància dels seus tipus respectius en algun instant t_1 anterior o igual a t .

Diem que p_i ($i=1, \dots, n$) és síncron en R si la participació d'una instància e_i de E_i en R en un instant t requereix que e_i sigui instància de E_i en t :

$$R(e_1, \dots, e_i, \dots, e_n, t) \rightarrow E_i(e_i, t)$$

Altrament, diem que es *asíncron*.

Un exemple de cada seria per *Escrit*(autor:Persona, Llibre). Si fos síncrona, si un autor morís i per tant no pertanyés a la població de Persona, tampoc existiria les instàncies de les relacions *Escrit* a les que participava. En canvi si fos *asíncrona*, si que podria donar-se el cas.

Durabilitat i Freqüència

En un sistema d'informació temporal, tot tipus d'entitat i relació té una durabilitat i una freqüència.

La durabilitat: pot prendre dos valors: instantani o durador. Diem que és instantani quan la seva vida té una duració igual a un instant de temps. En cas de durar més, diem que és durador. A continuació mostrem un exemple de cada, per un tipus de relació i un tipus d'entitat.

Tipus d'entitats:

- *instantani*: *PersonaEstornudadora*
- *durador*: *Planeta*

Tipus de relacions:

- *instantani*: *Vocalitza(Persona, Lletre)*
- *durador*: *Neix(Persona, Any)*

La freqüència també pot prendre dos valors: **únic i intermitent**. Diem que és únic, quan només té un interval de classificació. En cas de tenir més d'un diem que és intermitent. Un exemple de cada tipus seria.

Tipus d'entitat:

- *únic*: *Persona*
- *intermitent*: *Casat*

Tipus de relacions:

- *únic*: *Neix(Persona, Any)*
- *intermitent*: *Visita(Persona, Ciutat)*

En resum:

Freqüència	
Única	Intermitent
$E(e,t) \rightarrow Ci(e,E) = 1$	$E(e,t) \rightarrow Ci(e,E) \geq 1$
$R(e_1,...,e_n,t) \rightarrow Ci((e_1,...,e_n), R) = 1$	$R(e_1,...,e_n,t) \rightarrow Ci((e_1,...,e_n), R) \geq 1$

Taula 1 Resum formal de tipus de freqüències

Hi han articles que introdueixen com un nou tipus de durabilitat : constant i permanent, en les quals entrarem més a fons, però s'ha considerat que més bé seria una característica combinada amb una altra, la freqüència.

2.3 Restriccions d'integritat

2.3.1 Definició

Una **restricció d'integritat** és una condició que podria no ser satisfeta en alguns estats de la base d'informació o per events d'un sistema d'informació i per la qual s'han d'incloure mecanismes per satisfer-la en qualsevol instant, i així aconseguir les propietats de validesa i completesa.

A continuació definim les propietats de validesa i completesa:

- **Validesa**: Tots els fets de la base d'informació són certs.

- **Completesa:** La base d'informació conté tots els fets rellevants del domini.

Molt freqüentment diem que una base d'informació és íntegra, concepte derivat de la validesa i de la completesa. Cal remarcar que la integritat és una propietat molt important de les bases d'informació, ja que la manca d'integritat comporta normalment conseqüències negatives. Per tant, la base d'informació s'haurà de corregir en un moment o altre, i la correcció té un cost. A més, la falta d'integritat pot provocar que els usuaris poden prendre decisions incorrectes ja que parteixen d'informació que no és vàlida o que no és completa.

Una restricció pot ser simple o complexa. Una restricció és **complexa** si es pot descompondre en una conjunció d'altres restriccions; altrament, és **simple**. Per exemple, la restricció:

"Tots els empleats estan assignats a algun projecte"

és simple, perquè no la podem descompondre en un conjunt de restriccions. Una restricció com ara:

"Tots els empleats estan assignats a algun projecte, i tots els projectes tenen un nom"

seria complexa. Com que una restricció complexa es satisfà si, i només si, totes les seves restriccions components es satisfan, normalment només considerem restriccions simples.

Diem que una base d'informació és **consistent** si satisfà totes les restriccions d'integritat definides a l'esquema. També diem que una restricció d'integritat és **violada** quan no es satisfà en la base d'informació.

Una condició qualsevol sobre la base d'informació pot ser una restricció d'integritat si és una condició que satisfan les entitats i relacions del domini que estan representades a la base d'informació. Per tant, les restriccions d'integritat són representacions, en el sistema d'informació, de condicions que es satisfan en el domini, de la mateixa manera que, per exemple, la base d'informació representa les entitats del domini.

2.3.2 Causes de violació

Una restricció d'integritat és una condició que pot ser violada. La violació pot ser causada per:

- **L'arribada d'un cert missatge.** Per exemple :

"Una persona no pot obrir dos o més comptes el mateix dia".

Aquesta restricció es viola si una persona obre un segon compte en un mateix dia.

- **L'absència d'un o més missatges durant un cert interval temporal.** Per exemple:

"Tots els nous empleats són assignats a algun projecte abans de 30 dies".

Suposem que es contracta un empleat el dia d . Llavors, el sistema ha de rebre un missatge d'entrada, abans del dia $d+30$, indicant l'assignació de l'empleat a algun projecte. L'absència d'aquest missatge viola la restricció.

Quan una restricció d'integritat es viola, el sistema ha de produir alguna resposta per tal de mantenir la consistència de la base d'informació. En general, es poden distingir **tres menes de resposta**:

1. **Refusar** el missatge d'entrada que causa la violació. Aquesta resposta no és aplicable quan la violació és deguda a absència de missatge.
2. Executar una **acció compensatòria**. Això correspon a estendre el missatge d'entrada amb nous fets, de manera que el missatge estès manté la consistència. Si la violació és deguda a l'absència de missatge, aquesta resposta correspon a generar-ne un.
3. Demanar la **correcció** d'algun missatge anterior. Aquesta resposta és aplicable quan la violació és causada per algun missatge previ erroni, o per la no-presència en el seu moment d'algun missatge.

També són possibles altres respostes més elaborades, tals com marcar el missatge d'entrada com una excepció, o modificar la restricció d'integritat per a encabir l'excepció.

2.3.3 Tipus de Classificació

Les restriccions d'integritat es poden **classificar** segons diversos punts de vista. En general, en descrivim tres, segons:

1. L'**origen**, o el motiu pel qual la restricció s'ha de satisfer.
2. L'**abast**, o els elements del sistema d'informació als quals es refereix la restricció.
3. La **causa de la violació** de la restricció.

En aquest treball ens centrarem en la classificació segons l'abast, que té com a subtipus les restriccions d'integritat temporal o de transició.

Usualment, una restricció afecta només un conjunt limitat de fets i/o esdeveniments, i això permet classificar-la en funció dels fets afectats, o **abast**. Distingim sis classes: estàtica, **transició**, d'esdeveniment, d'història d'esdeveniments, global i precondition d'esdeveniment.

Una restricció d'integritat **estàtica** involucra només fets d'un estat de la base d'informació, i ha de ser satisfeta en tots els estats. Aquestes restriccions són tractades en tots els llenguatges de modelització conceptual.

Una restricció d'integritat de **transició** involucra fets de dos o més estats de la base d'informació. Usualment, només involucren fets de dos estats consecutius, restringint la transició entre ells, però en general la restricció pot referir-se a un nombre qualsevol d'estats. Per extensió, també s'anomenen de transició aquelles restriccions que afecten només *alguns* estats de la base d'informació. **Aquestes també reben el nom de restriccions dinàmiques o temporals.**

Una restricció d'integritat d'**esdeveniment** involucra només fets d'un sol esdeveniment.

Una restricció d'integritat d'**història d'esdeveniments** involucra dos o més esdeveniments, que ocorren en el mateix o diferent temps. S'usen sovint per expressar una ordenació temporal entre ocurrences d'esdeveniments.

Una restricció d'integritat **global** involucra fets de la base d'informació i dels esdeveniments. Les restriccions globals poden prendre qualsevol forma, però usualment involucren:

- dos o més esdeveniments i la base d'informació.
- un sol esdeveniment i la base d'informació en un estat posterior al de l'esdeveniment.
- la base d'informació en un cert estat i esdeveniments futurs.

Una restricció d'integritat de **precondició d'esdeveniment** és una classe particular de restricció global, que s'acostuma a considerar per separat perquè és molt important en modelització conceptual. Una restricció de precondició d'esdeveniment involucra un esdeveniment i els estats de la base d'informació fins abans de l'ocurrència de l'esdeveniment. Usualment, una restricció de precondició d'esdeveniment relaciona un esdeveniment i l'estat de la base d'informació anterior a l'ocurrència de l'esdeveniment, però en general, la restricció es pot referir a qualsevol estat anterior.

A continuació donem algunes restriccions i la seva classificació segons l'abast:

1. Tots els empleats estan assignats sempre a algun projecte (estàtica).
2. Una persona no pot ser titular de més de tres comptes en un cert instant (estàtica).
3. El salari d'un empleat no pot baixar (transició).
4. Un empleat no pot estar assignat a un projecte més d'un any (transició).
5. L'ingrés inicial, quan s'obre un compte, ha de ser superior a 1 euro (esdeveniment).
6. Una persona no pot obrir un compte si la suma dels saldos dels comptes que té és negativa (precondició d'esdeveniment).
7. Una persona no pot obrir dos comptes el mateix dia (història d'esdeveniments).
8. Tots els empleats fan un informe de les seves activitats almenys un cop al mes (global)

2.4 Classificació de restriccions d'integritat de transició

Ja s'ha vist que les restriccions de transició són condicions que involucren dos o més estats de la base d'informació. No són tan freqüents, ni han estat tan estudiades, com les estàtiques, però tot i així la major part dels sistemes d'informació tenen diverses restriccions de transició per controlar.

Pot semblar que, en general, les restriccions de transició s'han d'avaluar un cop sabem tots els fets de la base d'informació. Si fos així, les restriccions de transició tindrien poc valor pràctic, perquè hauríem d'endarrerir la seva avaluació al final del període de vida del sistema. En l'exemple:

“El salari d'un empleat no pot baixar”,

sembla que la restricció s'hagi d'avaluar un cop acabat el període de vida del sistema i coneguts. Per tant, seria massa tard per corregir les violacions que s'hagin pogut produir.

No obstant, s'ha fet la conjectura que la major part de les restriccions de transició es poden expressar en una de les dues formes que s'indiquen a continuació: universal i existencial. Quan és així, la restricció es pot avaluar en cada estat, involucrant només fets del propi estat i d'estats anteriors.

2.4.1 Restriccions de transició universal

Informalment, una restricció de transició **universal** defineix que en cada estat de la base d'informació, si una condició *CondEstat* és certa, llavors en *cadascun* dels estats anteriors t_i si és cert un *Antecedent* ha de ser cert també un *Conseqüent*.

La restricció "Els sous no poden baixar" és de transició universal perquè:

- *CondEstat* és "El sou d'un empleat e en t és s ".
- *Antecedent* és " e té un sou en t_i ".
- *Conseqüent* és "El sou de e en t_i és inferior o igual a s ".

2.4.2 Restriccions de transició existencial

Informalment, una restricció de transició **existencial** defineix que en cada estat de la base d'informació, si una condició *CondEstatActual* és certa, llavors ha d'existir almenys un estat anterior pel qual *CondEstatAnterior* és certa.

La restricció "Un jubilat ha d'haver treballat alguna vegada" és de transició existencial perquè:

- *CondEstatActual* és "Una persona p està jubilada en t ".
- *CondEstatAnterior* és "La persona p treballa".

De totes restriccions de transició, les més freqüents són si tracten amb:

2.4.3 Freqüència de tipus d'entitat i de relació

Ja hem definit en el punt anterior el concepte de freqüència. Per tant aquí s'englobaria el conjunt de restriccions que especifica un certa avaluació del valor per la freqüència per algun tipus d'entitat o de relació. Per exemple:

"Una persona pot estar més d'un cop casada"

El motiu d'aquest fet és per que pot estar divorciada o viuda un nombre indefinit vegades. Per tant $|Ci(e, Casat)| \geq 0$

2.4.4 Tipus d'entitat i de relació permanents

La característica d'ésser permanent té una estreta relació amb la durabilitat i la freqüència. Aquestes propietats ja han estat vistes en els punts anteriors. La idea principal és que en la població d'un tipus d'entitat o de relació les seves instàncies mai desapareixen. La definició donada no es del tot vàlida. Entrarem en profunditat en la seva definició en el capítol 3. Un exemple introductori seria:

"El sistema ha d'enregistrar tots els treballadors que s'ha contractat alguna vegada"

2.4.5 Tipus d'entitat i de relació constant

Aquesta característica és una especialització de les restriccions per tipus d'entitat i relació permanent. La idea general és que la població d'un tipus d'entitat o relació mai canvia, sempre és la mateixa. La definició donada no es del tot vàlida. Entrarem en profunditat en la seva definició en el capítol 3, tal i com hem fet en el punt anterior. Un exemple introductori seria:

"Cada casella d'un sudoku té una fila i una columna, sempre les mateixes"

2.4.6 Participació inicial d'un participant en un tipus de relació

La participació inicial és una característica d'un participant $p_i:E_i$, relativa a un tipus de relació $R(p_1:E_1, \dots, p_n:E_n)$ que defineix si les entitats de E_i han de participar i no en alguna relació de R , jugant el paper de p_i quan comencen un interval de classificació en E_i . Pot prendre dos valors: obligatòria o opcional.

Per simplificar la definició formal, suposarem que E és binari. L'ampliació al cas general és molt senzilla. Diem que la participació inicial de p_1 en $R(p_1:E_1, p_2:E_2)$ és **obligatòria** si quan una entitat e_1 comença un interval de classificació en E_1 ha de participar en alguna relació de R en aquell moment. Una altra manera de definir-ho és que la cardinalitat mínima donat e_1 en R ha de ser superior a 1.

Altrament diem que la seva participació és **opcional**. És a dir que la cardinalitat mínima, pot ser 0. Per exemple

HaEstatEn(Persona, País)

La participació de Persona és obligatòria, com a mínim ha estat en el País on es troba ara, en canvi la participació de País és opcional, pot no haver-hi estat cap persona.

2.4.7 Interval d'existència d'un tipus de relació en un participant

És una característica relativa a un tipus de relació $R(p_1:E_1, \dots, p_n:E_n)$ i un dels seus participants $p_i:E_i$, que defineix algunes característiques dels intervals durant els quals les entitats de E_i participen en alguna relació de R , jugant el paper p_i . Pot prendre tres valors: únic permanent, únic no permanent i múltiple.

Per simplificar la definició formal, suposarem que R és binari. L'ampliació al cas general és molt senzilla. Diem que l'interval d'existència de $R(p_1:E_1, p_2:E_2)$ en $p_1:E_1$ és **únic permanent** si a partir del moment que una entitat e_1 de E_1 participa en alguna relació de R , jugant el paper p_1 , llavors continua participant en alguna relació de R durant tot l'interval de classificació de e_1 en E_1 .

Per exemple, suposem el tipus de relació:

ÉsPropietat(Cotxe, propietària: Persona)

Si suposem també que a partir del moment que un cotxe té una persona propietària sempre més hi ha algun propietari, llavors l'interval d'existència de *ÉsPropietat* en cotxe és únic i permanent. La persona propietària podrà anar canviant, però sempre hi haurà algun propietari, mentre el cotxe existeixi.

Formalment l'interval d'existència de R en p_1 és únic permanent si

$$ic \in Ci(e_1, E_1) \wedge InclòsEn(t, ic) \wedge R(e_1, e_2, t) \wedge InclòsEn(t', ic) \wedge t' > t \rightarrow \exists e_2' (R(e_1, e_2', t'))$$

L'interval d'existència de R en p_1 és **únic no permanent** si, dins d'un interval de classificació t_i d'una entitat e_1 en E_1 , l'entitat e_1 participa en alguna relació de R durant un sol interval, que pot acabar abans que l'interval t_i .

Formalment es pot expressar així:

$$ic \in Ci(e_1, E_1) \wedge InclòsEn(t, ic) \wedge R(e_1, e_2, t) \wedge InclòsEn(t', ic) \wedge t' > t \wedge \neg \exists e_2' | R(e_1, e_2', t') \wedge InclòsEn(t'', ic) \wedge t'' > t' \rightarrow \neg \exists e_2'' (R(e_1, e_2'', t''))$$

Si l'interval d'existència de R en p_1 és únic permanent també és únic no permanent.
L'interval d'existència de R en p_1 és **múltiple** si no és únic no permanent.

2.4.8 Mutabilitat d'un tipus de relació respecte un participant

En un l'interval de d'existència de $R(p_1:E_1, \dots, p_n:E_n)$ en $p_i:E_i$ podem indicar que sempre pertanyi en les mateixes relacions o pugui canviar. Això és el que rep nom de mutabilitat. Els valors d'aquesta característica poden ser: mutable, immutable relatiu i immutable absolut. Un requisit per definir aquesta propietat és que el participant sota qui apliquem li apliquem un valor d'aquesta propietat ha de complir ser únic permanent.

Per simplificar la definició formal, suposarem que R és binari. L'ampliació al cas general és molt senzilla. Diem que $R(p_1:E_1, p_2:E_2)$ és **immutable relatiu** respecte de p_1 si:

$$ic \in Ci(e_1, E_1) \wedge InclòsEn(t, ic) \wedge R(e_1, e_2, t) \wedge InclòsEn(t', ic) \wedge t' > t \rightarrow R(e_1, e_2, t')$$

És a dir, un cop e_1 , dins d'un interval de classificació, participa en una relació de E, aquesta relació existeix fins al final d'aquell interval. Aquesta relació pot ser diferent d'un interval de classificació de e_1 a l'altre i, fins i tot pot ser que per algun interval no existeixi. Es pot precisar que sempre sigui la mateixa definint que R és **immutable absolut** respecte de p_1 . Formalment, això es dona si:

$$ic \in Ci(e_1, E_1) \wedge InclòsEn(t, ic) \wedge R(e_1, e_2, t) \wedge ic' \in Ci(e_1, E_1) \wedge InclòsEn(t', ic') \wedge t' > t \rightarrow R(e_1, e_2, t')$$

Si R no és immutable relatiu (i, per tant, tampoc absolut) respecte de p_1 , diem que és **mutable respecte de p_1** .

Per exemple:

Neix(Persona, llocNaixement: Ciutat)
Viu(Persona, llocResidència: Ciutat)

El tipus Neix és immutable absolut respecte de persona, perquè una persona no pot canviar de lloc de naixement. En canvi, el tipus Viu és mutable respecte de persona. Per altra banda, la característica de mutabilitat no és aplicable a Neix i Viu, respecte de ciutat, si els intervals d'existència d'aquests dos tipus en llocNaixement i llocResidència, respectivament, no compleixen el requisit esmentat (per exemple, si una ciutat pot deixar de tenir residents).

2.4.9 Inseribilitat en un tipus de relació respecte d'un participant

El propòsit de la característica de la inseribilitat en un tipus de relació R respecte d'un participant p_i , és indicar si és o no possible d'inserir noves relacions en R on participen instàncies de E, jugant el paper de p_i .

Això, igual que el cas anterior, només és aplicable per a participacions úniques permanents.

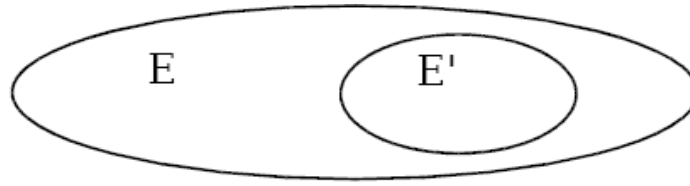


Figura 2 Gràfic representatiu de l'especialització

Per simplificar la definició formal, suposarem que R és binari. L'ampliació al cas general és molt senzilla. Diem que $R(p_1:E_1, p_2:E_2)$ és **no inserible** respecte de p_1 si:

$$ic \in Ci(e_1, E_1) \wedge InclòsEn(t, ic) \wedge R(e_1, e_2, t) \rightarrow R(e_1, e_2, Inici(ic))$$

En cas contrari diem que és **inserible**.

Per exemple, considerem el tipus de relació Pertany (LiniaComanda, Comanda). Aquest tipus de relació seria no inserible respecte de líniaComanda si, com és normal, la comanda d'una línia és inalterable. En canvi, podria ser inserible respecte de comanda si es permet que es puguin afegir línies a una comanda, un cop creada la comanda.

2.5 Les restriccions d'integritat temporals en les taxonomies

Les restriccions d'integritat temporal també afecten a les taxonomies indicant quins canvis de subtipus es poden realitzar dins d'aquestes. Per aquest motiu també entren en el nostre treball.

En aquest punt primerament es veurà el concepte d'especialització i generalització. A continuació s'explicaran les restriccions estàtiques associades a les taxonomies. Seguidament s'entrarà en les restriccions d'evolució de les particions, que no deixen de ser restriccions temporals, i finalment es definirà el concepte de *Valid Type Configuration Transition* i es parlarà sobre les reclassificacions.

2.1.1 Especialització i Generalització

Un tipus entitat E' és una especialització d'un tipus d'entitat E si E' té les propietats definides de E i altres de pròpies. Això també equival a dir que E és una generalització de E' . El tipus entitat E' rep el nom de **subtipus** de E i el de E de **supertipus** de E' .

Escrivim $E' \text{ IsA } E$ textualment per indicar que E' és una especialització de E , i $E \text{ Gens } E'$ per indicar que E és una generalització de E' , i així abreujar en quant a redacció.

En termes de poblacions, $E' \text{ IsA } E$ és una restricció d'inclusió de E' en E : en qualsevol moment les instàncies de E' són també instàncies de E . Naturalment, poden haver instàncies de E que no són de E' . Formalment, si $E' \text{ IsA } E$, llavors en lògica tenim:

$$E'(e) \rightarrow E(e)$$

La duració i la freqüència de les característiques de E' i E poden ser diferents, encara que sorgeixen dos restriccions:

- Si E és instantània, llavors E' també ha de ser instantània
- Si E és instantània i única, llavors E' ha de ser instantània i única.

2.1.2 Covering i Disjointness

Covering: Per indicar que qualsevol instància del supertipus és instància d'algun subtipus. Les generalitzacions que satisfan aquesta propietat s'anomenen **complete**, en cas contrari **incomplete**.

Formalment:

$$E(e) \rightarrow E_1(e) \vee \dots \vee E_n(e)$$

Disjointness: Per indicar que qualsevol instància d'un subtipus E_i no és de cap altre subtipus de la mateixa taxonomia. Les restriccions que satisfan aquesta propietat són anomenades **disjoint**, en altre cas **overlapping**.

Formalment:

$$E_i(e) \rightarrow \neg E_1(e) \wedge \dots \wedge \neg E_{i-1}(e) \wedge \neg E_{i+1}(e) \wedge \dots \wedge \neg E_n(e)$$

Les taxonomies que compleixen les restriccions de disjoint i covering reben el nom de **particions**.

Un tipus d'entitat donada pot ser el supertipus de vàries particions. Cada partició té una **dimensió** diferent, és a dir, que cadascuna té un **pseudoatribut** diferent per poder dur a terme la classificació de cada instància del supertipus en cadascun dels subtipus.

El concepte de partició pot ser bàsic o derivat. Aquí considerem que els conceptes derivats, poden ser

- Derivats per especialització
- Derivats per exclusió
- Derivats per unió

En molts casos, la derivació de conceptes pot comportar un conjunt de restriccions de particions. Una partició té un tipus d'evolució, la qual esta relacionada amb la classificació dinàmica, el qual marca unes restriccions (**specialization evolution constraints**), que és important mencionar.

2.1.3 Restriccions d'evolució d'especialitzacions

Un **tipus de configuració** d'una entitat e en un instant t , denotat per $TC(e,t)$, és el conjunt de tipus d'entitats $TC(e,t) = \{ E_i \mid e \text{ és instància de } E_i \text{ en } t \}$. Altres noms usats són **role set** i **context**.

Les restriccions d'evolució d'especialitzacions serveixen per determinar les evolucions vàlides del tipus de configuracions.

Existeixen tres restriccions de evolució d'especialització: absolutament estàtica, relativament estàtica i dinàmica, les quals caracteritzen la possible evolució de les poblacions d'un subtipus i un supertipus en una especialització, definint possibles canvis entre subtipus. Per a cada especialització, el dissenyador deuria definir quines de les tres millors restriccions corresponen a l'evolució de les poblacions del subtipus i del supertipus.

2.1.3.1 Absolutament estàtica (respecte un supertipus)

Donada una especialització, un subtipus E' és absolutament estàtic respecte un supertipus E , si totes les entitats/relacions que són instàncies del subtipus, no poden ser en cap moment, ni han estat en un moment anterior, instàncies del supertipus E sense ser-ho de E' . (D'una forma més simplificada, no es permet classificació dinàmica dins d'una partició).

La definició formal és:

$$\forall e(\exists t \mid E'(e,t) \rightarrow \forall t' (E(e,t') \rightarrow E'(e,t'))), \text{ que és equivalent a: } \\ \forall e(\exists t \mid E'(e,t) \rightarrow Ci(e,E') = Ci(e,E))$$

Un exemple seria: En un joc de mots encreuats, les caselles negres és absolutament estàtic de casella: si una casella és instància de negra, en un instant $t=10$, llavors no és possible que en un altre instant sigui instància de casella sense ser-ho de negra.

2.1.3.2 Relativament estàtica (respecte un supertipus)

Si donada una especialització, totes les instàncies que en algun moment són d'un subtipus E' , i que per tant, tenen un interval de classificació ci en el supertipus E , no poden ser en algun altre **moment inclòs en ci** , instàncies de E sense ser-ho de E' . Com per exemple

$$P = \text{NoCasat} \textbf{Partid} \text{Solter, Divorciat, Viduu}$$

Per tal de deixar de ser solter ha de deixar de ser no casat, ja que en cas de que hagi de passar a divorciat o a vidu, primer s'ha de casar i per tant deixa de pertànyer a no casat.

2.1.3.3 Dinàmica (respecte un supertipus)

Si donada una especialització, no hi ha restriccions afegides. Per exemple, un cotxe pot estar en ús o en reparació, i pot saltar a formar part d'un tipus d'entitat a un altre indistintament.

En les particions relatives estàtiques i dinàmiques, és freqüent el cas que hi hagi una manera en que les instàncies puguin canviar el seu subtipus. En aquest casos, les transicions admissibles són usualment definides mitjançant diagrames d'estat, els quals descriuen els estats inicials permesos, les transicions vàlides, i els estats finals permesos.

En general, les restriccions de evolució d'una especialització entre un subtipus i un supertipus necessàriament no és la mateixa per tots els subtipus de la partició. Nosaltres diem que la partició és híbrida si la restricció no és la mateixa per tots els subtipus, altrament és pura. Cada partició híbrida pot ser transformada conservant la informació en una pura introduint tipus d'entitats auxiliars i particions però no és l'objectiu d'aquest document. Per exemple:

$$\text{Llibre} \textbf{Partid} \text{LlibreNoLlogable, LlibreDeLloguer, LlibreDisponible}$$

en el que LlibreNoLlogable és absolutament estàtica de Llibre , i les altres dos són dinàmiques

2.1.4 Valid Type Configuration Transitions

Quan una entitat és creada, aquesta té un tipus de configuració - *type configuration* -, la qual ha de ser vàlida com a inicial.

Des d'aquest moment, la entitat pot evolucionar i canviar el seu tipus de configuració. En qualsevol moment, l'entitat tindrà algun tipus de configuració la qual serà buida si la entitat és 'suspended'. Es defineix com a **història de configuracions** tipus d'una entitat e en el temps t , $H(e,t)$, com la seqüència de tipus de configuracions des del seu instant de creació fins a t .

$$H(e,t) = \langle TC(e,t_0), TC(e, t_0 + 1), \dots, TC(e,t) \rangle$$

on t_0 és el instant de creació de e . Tots els $TC(e, t_0 + 1), \dots, TC(e, t-1), TC(e,t) >$ són tipus de configuracions vàlides. D'altra banda dos configuracions consecutives poden ser les mateixes si e no ha canviat d'un instant a l'altre.

Una història $H(e,t)$ és vàlida si:

1. La vida de e en cadascun dels tipus d'entitats incloses en algun $TC(e,t_i)$ satisfà les característiques temporals associades i
2. $H(e,t)$ satisfà la especialització i la partició de les partition evolution constraints.

Una transició $Tr(e,t)$, $t > t_0$ és el parell $\langle H(e,t-1), TC(e,t) \rangle$. La evolució per tant d'una entitat és una seqüència de transicions. Una vegada una transició pren part, la nova història és:

$$H(e,t) = H(e,t-1) + \langle TC(e,t) \rangle$$

on $+$ denota l'operació de concatenació. Diem que la transició és vàlida si són vàlides $H(e,t-1)$ i $H(e,t)$. També diem que una història és vàlida si totes les seves transicions són vàlides.

Hi han els següents tipus de **transicions genèriques**:

- **Reclassificació:** Quan dos tipus de configuracions consecutives d'una mateixa entitat no són buides i són diferents
- **Continuació activa:** Quan dos tipus de configuracions consecutives d'una mateixa entitat no són buides i són iguals
- **Suspensió:** Quan donats dos tipus de configuracions consecutives, la primera és no buida i la segona és buida
- **Continuació de suspensió:** Quan dos tipus de configuracions consecutives d'una mateixa entitat són buides
- **Reactivació:** Quan donats dos tipus de configuracions consecutives, la primera és buida i la segona no és buida.

Llavors, donat el cas de les restriccions dinàmiques constant i permanent aplicades a particions, creiem que la transició més destacada a estudiar seria la de reclassificació.

Reclassificacions

Donada una transició $Tr = \langle \langle TC_0, \dots, TC_{n-1} \rangle, TC_n \rangle$, on Tr és una transició de reclassificació genèrica. Tr és vàlida si satisfà les 5 condicions següents.

- Si $E \in TC_{n-1}$ i E és instantània i única, llavors E no pertany a TC_n .
- Si hi ha una partició absoluta o relativament estàtica $P = E \text{ Partd } E_1, \dots, E_i, \dots, E_n$
 - E i $E_i \in TC_{n-1}$, i
 - $E \in TC_n$llavors $E_i \in TC_n$
- Si hi ha una partició dinàmica $P = E \text{ Partd } E_1, \dots, E_i, \dots, E_j, \dots, E_n$
 - E i $E_i \in TC_{n-1}$, i
 - E i $E_j \in TC_n$llavors, E_j pertany als possibles subtipus successor de E_i , P
- Si hi ha una partició relativament estàtica o dinàmica $P = E \text{ Partd } E_1, \dots, E_i, \dots, E_j, \dots, E_n$
 - Existeix una k , $0 \leq k < n-1$, tal que E i $E_i \in TC_k$, i
 - No existeix una s , $k < s < n$, tal que $E \in TC_s$, i
 - E i $E_j \in TC_n$.llavors E_j pertany als possibles subtipus finals de E_i en P
- Si hi ha una partició relativament estàtica o dinàmica $P = E \text{ Partd } E_1, \dots, E_i, \dots, E_n$
 - $E, E_i \in TC_n$, i
 - No existeix una k , $0 \leq k < n$, tals que $E \in TC_k$llavors E_i pertany als inicials subtipus de P .

2.2 Restriccions d'integritat temporal amb lògica temporal

2.2.1 Introducció

Una manera alternativa d'expressar les restriccions de transició és mitjançant la lògica temporal, que és una extensió de la lògica de primer ordre, amb operadors temporals. Existeixen diferents variants segons els operadors que defineixin encara que hi ha un conjunt establerts universalment.

Varies característiques de la lògica temporal la fan especialment atractiva com llenguatge per definir consultes i restriccions d'integritat en base de dades temporals. Per tant, també serveix per treballar amb bases d'informació històriques.

En lògica temporal, es considera que la base d'informació passa per una seqüència d'estats $\langle \sigma_0, \dots, \sigma_n \rangle$ que conforma la seva història. Un estat de la base d'informació conté els fets d'entitats i de relacions certs en aquell estat. La representació lògica d'aquests fets no inclou ara l'argument temporal, perquè és implícit: els fets són certs en un estat determinat. Així, en comptes d'un predicat *Sou(persona, import, temps)* tindríem ara el predicat *Sou(persona, import)*, sense el temps.

Es pot considerar que hi ha un estat per cada instant del període de vida del sistema, però no és obligatori. De fet, és més freqüent suposar que un estat correspon a un interval del període de vida del sistema, durant el qual són certs els mateixos fets. Cada cop que canvia un o més fets, hi ha un canvi d'estat, de σ_i a σ_{i+1} .

2.2.2 Història de la lògica temporal

En l'àrea de la lògica va sorgir la necessitat d'escriure expressions on intervén el temps. Aquest problema es va solucionar primerament mitjançant la lògica de primer ordre afegint a tots els predicats un paràmetre addicional 't' que representava un instant de temps. Hi ha escrits que ja això ho consideren una lògica temporal. Aquesta extensió de la lògica de primer ordre rep el nom de FOL2.

Posteriorment, es va aplicar com a solució crear un tipus de lògica que ja oferís connectors amb semàntica temporal sense fer explícit el paràmetre 't' que usa FOL2. Aquesta proposta va ser el que va originar la lògica temporal, de la qual hi ha moltes varietats en funció dels operadors que ofereixen, com ja s'ha comentat anteriorment.

Generalment, a mesura que apareix una lògica, aquesta forma un llenguatge que inclou el llenguatge d'una lògica anterior. Això és el que passa per exemple amb la lògica de primer ordre i la lògica proposicional.

Lògica proposicional \subset Lògica de primer ordre

Per tant, és interessant estudiar la lògica temporal a partir del conjunt de lògiques anteriors per conèixer completament el seu llenguatge. Hi ha les següents lògiques:

2.2.2.1 Lògica proposicional

La lògica temporal es caracteritza per estudiar les proposicions o sentències lògiques, les seves possibles avaluacions de veritat, i en cas ideal, el seu nivell absolut de veritat.

Una proposició és un enunciat declaratiu que pot ser cert o fals, però no els dos alhora.

Les proposicions es representen mitjançant variables proposicionals simbolitzades amb lletres. Amb la combinació de variables proposicionals i conjuncions, definides com funcions de veritat, s'obtenen fórmules sentencials o sentències.

Els símbols bàsics que constitueixen l'alfabet és:

- Símbols de veritat: per a cert i \perp per a fals.
- Símbols de variables: p, q, r, s, ...
- Símbols de connectives: \neg , \wedge , \vee , \rightarrow , \leftrightarrow
- Símbols de puntuació: (,), per evitar ambigüitats.

Un exemple:

$$p \wedge q \rightarrow \neg(\neg p \vee \neg q)$$

2.2.2.2 Lògica de primer ordre (FOL)

La lògica de primer ordre extén de la lògica proposicional i que es caracteritza principalment de que el seu vocabulari conté quantificadors de variables.

Existeix un domini **D**, que és el que s'anomena "Domini de discurs": el conjunt d'objectes dels quals parlem. Si volem parlar, donem per fet que hi ha d'haver alguna cosa de la qual parlar, llavors el domini de discurs sempre té almenys un objecte, però pot ser finit o infinit. Per tant, és la base de la qual les fórmules són construïdes.

Les expressions de FOL es defineixen a partir d'un conjunt de símbols, el vocabulari, format per les connectives:

- Les **connectives proposicionals**, $\{ \neg, \wedge, \vee, \rightarrow, \leftrightarrow, \perp, \top \}$
- Els **quantificadors**, $\{ \forall, \exists \}$

, un conjunt de variables x, y, z, \dots (són variables sobre els elements del domini de discurs) i el vocabulari no lògic, que és un conjunt de:

- **constants** a, b, c, \dots ; una constant s'interpreta com un element del domini del discurs;
- **lletres de funció** ("funcions") f, g, h, \dots , un conjunt per a cada arietat $n > 0$; (n -àries): Una lletra de funció s'interpreta com una operació n -ària sobre el domini de discurs.
- **Lletres proposicionals** P, Q, R, \dots : Una lletra proposicional s'interpreta com $\{ \langle \rangle \}$ o $\{ \}$
- **Lletres de predicat** ("predicats") P, Q, R, \dots , un conjunt per a cada arietat $n > 0$: Una lletra de predicat n -ària s'interpreta com una relació n -ària sobre el domini de discurs.

Com podem veure tant les lletres proposicionals com les connectives proposicionals, ja pertanyien a la lògica proposicional.

En lògica de primer ordre hi ha dos tipus d'expressions:

- **Els termes**: són les que tenen valors "de referència", és a dir, elements del domini del discurs (equivalentment, operacions 0àries).
 - Una variable és un terme: pronom per algun element del domini.
 - Una constant és un terme: nom per algun element del domini
 - Si f és una funció d'arietat n i t_1, \dots, t_n són termes, llavors f és un terme: el resultat d'aplicar l'operació representada per f als objectes representats per t_1, \dots, t_n
- **Les fórmules**: que contenen valors "de veritat": o bé $\{ \langle \rangle \}$, que llegim cert, o abreujadament 1, o bé $\{ \}$, que llegim "fals" i abreugem 0 (és a dir les relacions 0àries).

Si P és una lletra proposicional, P és una fórmula, ja que és llegeix cert o fals.

- Si P és un predicat d'arietat n i t_1, \dots, t_n són termes, llavors P és una fórmula, on el sentit semàntic és que els objectes representats per t_1, \dots, t_n compleixen la relació representada per P .

Aquestes reben el nom de **fórmules atòmiques**

- Cert és una fórmula
- Fals és una fórmula
- Si A és una fórmula, $\neg A$ és una fórmula
- Si A i B són fórmules $A \wedge B$ és una fórmula
- Si A i B són fórmules, $A \vee B$ és una fórmula
- Si A i B són fórmules $A \rightarrow B$ és una fórmula

Aquestes juntament amb les anteriors formen les **fórmules proposicionals**.

- Si A és una fórmula, i x és una variable, $\exists xA$ és una fórmula: la qual cosa es llegeix que hi ha un objecte x del domini tal que és que A (quantificació existencial).
- Si A és una fórmula i x és una variable, $\forall xA$ és una fórmula: la qual cosa es llegeix que per a tot objecte x del domini és que A (quantificació universal).

El conjunt de totes les fórmules definides són **les fórmules de primer ordre** o simplement les fórmules.

Un enunciat o fórmula tancada, és una fórmula que no té cap variable lliure, és a dir, cap variable quantificada. En particular, el valor d'un enunciat ha de ser el mateix per a tota assignació.

D'altra banda, el valor d'una fórmula és independent de l'assignació a variables que no hi apareixen lliures. Per tant depèn de les lliures.

Notació

Per facilitar la lectura de la notació textual assumim una abreviatura segons la convenció que les connectives unàries ($\neg, \forall x, \exists x$) tenen precedència més alta que les connectives binàries, i que \wedge i \vee tenen precedència més alta que \rightarrow .

“2FOL”

Els valors de les fórmules en lògica clàssica són independents del temps. **Llavors per poder obtenir valors de veritat diferents en funció de l'instant en que s'avalua l'expressió, no queda més remei que incorporar el temps com un element del domini, i que aquest sigui modelat com una variable explícita en la fórmula.**

Alguns documents la classifiquen com a una lògica temporal ja, encara que no tingui operadors amb semàntica temporal implícita. En aquests documents rep el nom de “*Two sorted First order logic*” o abreviat “**2FOL**”

Per exemple, en modelització conceptual, indicariem que una instància e es de la entitat tipus E en el instant t mitjançant un predicat amb el mateix nom que el nom del tipus d'entitat i amb dos temes, els quals serien dos variables que representen dos objectes del domini: e i t. $E(e,t)$.

Per tant la lògica de primer ordre ens permet encara definir aspectes temporals, però al representar més elements dins de la fórmula, fa que és converteixi més farragosa i deixi de ser el suficientment expressiva.

Tornant a l'exemple del punt de la introducció: El sou d'un empleat ha de ser inferior al del seu cap, es formularia així:

$$\begin{aligned} TeSou(x,y) &\rightarrow Empleat(x) \wedge Diners(y) \\ TeSou(x,y) \wedge TeCap(x,z) &\wedge TeSou(z,w) \rightarrow y < w \end{aligned}$$

Però com formular que el seu sou no pot decreixer? Aplicant 2FOL ens quedaria (**suposant que empleat és permanent**):

$$TeSou(x,y,t) \rightarrow TeSou(x,w,t+1) \wedge y \leq w$$

2.2.2.3 Lògica d'alt ordre (HOL)

La lògica d'alt ordre, HOL, estén la domini modelat per FOL permetent l'adopció d'aplicar als predicats una quantificació, com si es tractes de variables. Per exemple

$\forall P / \exists x / P(x)$,

no pot ser escrit en FOL perquè conté un quantificador que fa variar un predicat P.

2.2.2.4 Lògica modal

La lògica modal és la mare de les lògiques temporals i es la que introdueix el concepte de món, com element que fa variar si una fórmula es certa o falsa, és a dir, els conceptes de veritat o falsedat no són estàtics i immutables, però són, en sentit contrari, relatius i variables.

Posteriorment aquest concepte de món s'aplica en lògica temporal per definir un instant de temps o més ben dit, el que correspondria a un estat de la base d'informació. Un exemple seria:

Possiblement Joan aprovi l'examen

Per tant, la lògica modal intenta apropar-se més al pensament humà i del llenguatge natural, complementant la lògica de predicats amb modalitats que indiquen les condicions en las que és certa o falsa cada fórmula.

Per exemple, afegeix paraules com pot ser, podria, ha de, possiblement, necessàriament, a vegades, etc. Els operadors afegits són de L (necessàriament) i M (possiblement):

M aprova(Joan, examen)

2.2.2.5 Lògica temporal

Les lògiques temporals són un model particular de lògica modal, on el conjunt de mons (estats en els que es satisfà la fórmula) són interpretats com un conjunt de tots els possibles instants T d'un domini temporal. D'acord amb la introducció, podem passar a interpretar T com el conjunt d'estats d'una base d'informació.

Generalment la lògica temporal afegeix **4 nous operadors universals** a la lògica proposicional o FOL que oculten les quantificacions en el domini temporal.

G, sempre en el futur: \Box F, eventualment en el futur \Diamond H sempre en el passat \blacksquare P eventualment en el passat \blacklozenge
--

En la definició dels operadors s'usa la notació $V(F, t)$, on:

- F és la fórmula que s'avalua
- t és l'instant/estat que es pren com a referència per seleccionar els instants/estats amb els que avaluar F.

Per la validació d'una restricció d'integritat, escrita amb una fórmula F que conté operadors de lògica temporal, es substituirà t per cada estat de la base d'informació i tornarà a avaluar-se F. A continuació veiem la definició dels operadors de lògica temporal:

$V(\Box f, t) = \text{cert si } \forall s (s \in T \wedge t < s \rightarrow V(f, s))$

Patró de lectura: Aquell que sempre es satisfarà $\langle f \rangle$.

Exemple: $\Box \text{Treballa}(x, y)$

Aquell que sempre es satisfarà treballa.

$V(\Diamond f, t) = \text{cert si } \exists s (s \in T \wedge t < s \rightarrow V(f, s))$

Patró de lectura: Aquell que satisfarà algun cop $\langle f \rangle$

Exemple: $\Diamond \text{Ascendeix}(x)$
Aquell que satisfarà algun cop ascendeix.

$V(\blacksquare f, t) = \text{cert si } \forall s (s \in T \wedge s < t \rightarrow V(f, s))$
 Patró de lectura: Aquell que sempre ha satisfet $\langle f \rangle$
 Exemple: $\blacksquare \neg \text{Orfe}(x)$
Aquell que sempre ha satisfet no orfe.

$V(\blacklozenge f, t) = \text{cert si } \exists s (s \in T \wedge t < s \rightarrow V(f, s))$
 Patró de lectura: Aquell que ha satisfet algun cop $\langle f \rangle$
 Exemple: $\blacklozenge \text{Empleat}(x, \text{FIB})$
Aquell que ha satisfet algun cop empleat de la FIB

Si la relació d'ordre($<$) és transitiva i no reflexiva, generalment és així, és possible introduir dos operadors binaris més:

$V(f1 \text{ until } f2, t) = \text{cert si } \exists s (s \in T \wedge t < s \wedge V(f2, s) \wedge \forall u (u \in T \wedge t < u < s \rightarrow V(f1, u)))$
 Patró de lectura: Aquell que satisfà contínuament $\langle f1 \rangle$ fins que satisfà $\langle f2 \rangle$
 Exemple: $\text{Matriculat}(x, \text{FIB}) \text{ until } \text{Graduat}(x, \text{Informàtica})$
Aquell que satisfà contínuament matriculat a la FIB fins que satisfà graduat en Informàtica.

$V(f1 \text{ since } f2, t) = \text{cert si } \exists s (s \in T \wedge t > s \wedge V(f2, s) \wedge \forall u (u \in T \wedge t > u > s \rightarrow V(f1, u)))$
 Patró de lectura: Aquell que satisfarà contínuament $\langle f1 \rangle$ des de que satisfà $\langle f2 \rangle$
 Exemple: $\text{Treballa}(x, \text{IBM}) \text{ since } \text{Graduació}(x, \text{Informàtica})$
Aquell que satisfarà contínuament treballa en IBM des de que satisfà graduació en informàtica.

Altres operadors comuns són next o prev representats per \circ i \bullet respectivament. Aquests operadors són unaris i poden ser definits en termes de until i since:

$\circ f1 \equiv \text{fals until } f1$
 Patró de lectura: Aquell que el següent instant compleix $\langle f1 \rangle$
 Exemple: $\circ(\text{ComençaClasse}(y) \rightarrow \text{Aula}(x, y))$
Aquell que el següent instant compleix que si comença classe, hi haurà aula

$\bullet f1 \equiv \text{fals since } f1$
 Patró de lectura: Aquell que l'anterior instant compleix que $\langle f1 \rangle$
 Exemple: $\bullet \text{CursaMaster}(x, \text{ESADE}) \wedge \text{Treballa}(x, \text{ESADE})$
Aquell que l'anterior instant compleix CursaMaster en ESADE i que satisfarà algun cop treballa en ESADE

Ara docs, faria falta estendre la definició de fórmula:

Si A és una fórmula, $\Box A$ és una fórmula
 Si A és una fórmula, $\Diamond A$ és una fórmula
 Si A és una fórmula, $\blacksquare A$ és una fórmula
 Si A és una fórmula, $\blacklozenge A$ és una fórmula
 Si A i B són fórmules, A **until** B és una fórmula
 Si A i B són fórmules, A **since** B és una fórmula
 Si A és una fórmula, **until** A és una fórmula
 Si A és una fórmula **since** A és una fórmula

Un cop sabem això, ara podríem formular l'exemple de que el sou d'un empleat no pot decreixer, que quedaria:

$$\text{TeSou}(x, y) \rightarrow \Box \text{TeSou}(x, w) \wedge y \leq w$$

La forma de computar mentalment això seria obtenir les x , y del predicat “TeSou” que és cert per algun estat de la base d’informació, (o el que equivaldria en 2FOL passar per tots els instants de temps), i mirar si per tots els estats futurs el predicat TéSou es satisfà pels valors de x de l’estat anterior i si y (el sou anterior) no és superior al de l’estat més futur.

2.2.3 Lògica temporal: Modelat del Temps

A vegades, en una expressió lògica ens serà necessari parlar d’instàncies concretes de temps. El temps té propietats que poden tenir diferents valors en funció de com el vulguem tractar.

A continuació presentem quines propietats el modelen i quins valors hauríem com a mínim de definir.

El volum

Pot tenir dos valors: **discret** o **dens**. El volum del temps és discret si les seves unitats corresponguin a enters. En cas de que les seves unitats corresponguin a racionals o reals estarem parlant d’un temps amb un volum dens.

L’ontologia

Té dos valors, per **punts** (instants) o per **intervals** (períodes). Aquesta és una de les propietats més tractades sobre tot en textos que tracten de bases de dades temporals. El temps és tractat per punts, si cada fet és cert pels instants de temps marcats. En cas de que sigui cert des d’un instant de temps fins a un altre, direm que el temps es tracta com a intervals.

Punt de referència

Moltes de les cerques per a dates, s’ha enfocat en modelar-lo amb **temps absolut** d’alguna manera; això és, punts exactes de temps o intervals associats a cada dada. Per exemple:

TeSalari('Joan', 1200€, 1/5/200)

En canvi, en IA, s’ha donat gran atenció en el emplear el **temps relatiu**, per permetre per exemple, modelització d’expressions com la setmana passada, fa un any,..., les quals apareixen en el llenguatge natural per entendre la història dels sistemes.

Dimensionalitat

El temps és **multidimensional** quan tractem diferents tipus de temps (per exemple temps de transacció (instant de temps real en el que un predicat és cert) i temps vàlid (instant de temps en el que un predicat consta en la base d’informació conforme és cert)). En cas contrari és **unidimensional**.

Linealitat

Es diu que parlem de temps **no lineal**, quan hi ha la necessitat de fer assercions de futurs que existiran alhora, en cas contrari, parlem de temps lineal. Per tant, si la estructura del temps és **lineal** un estat del sistema pot ser associat a cada instant de temps.

2.2.4 Classificacions de les lògiques temporals

En lògica temporal, en cas que vulguem tractar un interval de temps concret, si ja no tenim el paràmetre temps de 2-FOL possiblement haurem de trobar un operador que tracti de forma equivalent aquest interval de temps. Cal ressaltar, que està demostrat el següent:

l'expressivitat de 2-FOL \supset l'expressivitat de la lògica temporal.

Degut a la varietat de maneres que hi ha de tractar el temps, no pot haver una única existència d'una única lògica temporal. Hi han varies, d'aquí l'aparició d'una llarga llista de lògiques temporals: *PTL*, *Choopy Lògic*, *BTTL*, *ITL*, *PMLTI*, *CTL*, *IL*, *EIL*, *RTIL*, *LTI*, *RTTL*, *TPTL*, *RTL*, *TRIO*, *MTL*, *TILCO*...

El conjunt de lògiques temporals presenten unes característiques particulars per classificar-les, les quals veurem a continuació. Algunes d'aquestes característiques coincideixen notablement amb les propietats de modelatge del temps, i són en les que ens centrarem estudiant com les tracten:

Definició de les propietats de la relació entre temps

En un domini on té influència el temps, hi ha una relació entre cada estat emmarcat dins d'un instant de temps i un altre.

Per tant hi ha una relació R entre els instants de temps, representada per $<$. Aquesta relació equivaldria a la relació entre mons vista en la lògica modal, i té unes propietats a seleccionar en funció del domini que tractem:

Transivitat $\forall x,y,z(x < y \wedge y < z \rightarrow x < z)$
Noreflexivitat $\forall x(\neg x < x)$
Linealitat $\forall xy(x < y \vee x = y \vee y < x)$
Linealitat esquerra $\forall xyz(y < x \wedge z < x \rightarrow y < z \vee y = z \vee z < y)$
Linealitat dreta $\forall xyz(x < y \wedge x < z \rightarrow y < z \vee y = z \vee z < y)$
Començament $\exists x(\neg \exists y(y < x))$
Final $\exists x(\neg \exists y(x < y))$
Predecessor $\forall x(\exists y(y < x))$
Successor $\forall x(\exists y(x < y))$
Densitat $\forall xy(x < y \rightarrow \exists z(x < z < y))$
Discretitat $\forall xy(x < y) \rightarrow \exists z(x < z \wedge \neg \exists u(x < u < z)) \wedge \forall xy(x < y \rightarrow \exists z(z < y \wedge \neg \exists u(z < u < y)))$

Tractament del temps per punts o per intervals

Suposant que un fet pot ser cert en diferents punts/intervals del temps, ha d'haver alguna manera de indicar-ho, i sol ser útil per definir 'en quin temps' tracten les restriccions d'integritat.

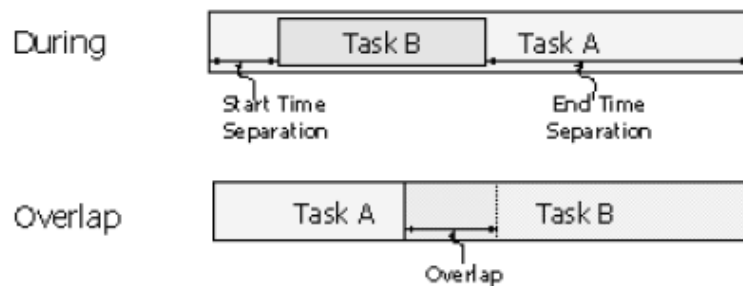


Figura 3 Exemple de lògica basada en intervals

Exemple:

Abans del dia 9 de juny del 2007 s'ha d'haver registrat la nova versió de cada COT.

Hi han dos maneres concretament, mitjançant:

- **La lògica basada en punts:** que permet expressar les relacions de fets basats punts.
- **La lògica basada en intervals:** descriu les relacions entre intervals. Principalment destaca per permetre escriure fórmules amb un nivell més alt d'abstracció, amb operadors específics, essent més concises i fàcils d'entendre que les lògiques basades en punts.

Un exemple d'operadors serien: *before*, *meets*, *during*, *overlap*.. tal i com es veu a la figura següent:

Restriccions temporals quantitatives o qualitatives (mètrica)

Hi han altres tipus de restriccions que depenen de si hi ha una mètrica o no :

- **Restriccions temporals quantitatives:** permeten expressar condicions sobre l' inici i duració de la satisfacció d'una fórmula respecte l'instant d'avaluació (en cas d'usar mètrica). Un exemple:

[4,7] A: 4 unitats de temps de distància respecte l'instant d'avaluació i 7 de duració.

En cas de que s'indiqui un sol valor, s'estarà referint a la distància en que la fórmula es satisfà:

≤ 5 A: 5 unitats de temps de distància.

No a tots els operadors temporals els hi poden afegir aquesta quantificació, degut a la semàntica, ja que aquests valors són interpretats relatius a l'instant d'avaluació de la fórmula. Per tant només aquells que tractin amb el futur podran usar-los (\square , \diamond).

La semàntica dels límits en un fórmula, pot ser entre **claudàtors** o **entre parèntesi**, amb la mateixa interpretació que en les matemàtiques: claudàtor indica límit tancat, i parèntesi límit obert.

Una altra manera de quantificar el temps és emprar un **quantificador temporal** amb una variable sota el seu abast de la mateixa manera que usàrem els quantificadors de FOL. Per exemple:

$$\Box x (p \rightarrow (\Diamond y (q \wedge y \leq x+6)))$$

Això significa que per cada estat del temps x , si p es satisfà, llavors hi ha un estat futur amb el temps y tal que ' q ' es satisfà i ' y ' és com a molt $x+6$.

- **Restriccions temporals qualitatives:** permeten expressar les relacions entre fets de causa-conseqüència de satisfacció de fórmules, descrivint d'aquesta manera l'evolució del sistema. Un exemple: Els empleats es formen abans de treballar

$$Empleat(x) \rightarrow (Forma(x) \textbf{ precedes } Treballa(x))$$

Els operadors característics són els següents:

$A \textbf{ precedes } B$

$A \textbf{ follows } B$

A serà cert fins B sigui cert la pròxima vegada (until)

A ha estat cert des de l'última vegada que B va ser cert l'últim cop.(since)

Totes aquestes expressions però, poden ser definides amb els operadors ja vistos de until i since. Els dos primers:

$A \textbf{ precedes } B = \neg ((\neg A) \textbf{ until } B)$

$A \textbf{ follows } B = \neg ((\neg A) \textbf{ since } B)$

Els dos últims per la pròpia definició de until i since, encara que hi ha lògiques que ofereixen diverses definicions d'aquests operadors.

Tractament del temps de forma implícita, explícita o absoluta

El temps en lògica temporal pot ser **declarat**/definit d'una manera:

- **Implícita** : Quan la fórmula depèn del temps d'avaluació. El temps és implícit a la part esquerra de la fórmula. Per exemple:

$$\Box A.$$

- **Explícita**: La representació es fa a través d'una variable. D'aquesta manera és possible expressar qualsevol cosa. Per exemple:

$$\forall t (\Box (E \wedge T = t) \rightarrow (A \wedge (T - t) < 10ms))$$

La **referència** del temps pot ser:

- **Absoluta** : El valor del temps ve donat pel valor d'un sistema de rellotge, generalment representat per la variable T . En aquest cas, les duracions venen donades directament en la mesura absoluta marcada pel rellotge, com per exemple segons. Per exemple:

$$\forall t(\Box (E \wedge T = t) \rightarrow (A \wedge (T - t) < 10ms))$$

- **Relativa**: El temps és expressat d'una manera relativa, és a dir, les duracions són donades en unitats de temps. La relació entre el temps absolut i l'expressió, es deixa per la fase d'implementació. Per exemple:

$$\forall t(\Box (E \wedge T = t) \rightarrow (A \wedge (T - t) < 10))$$

2.3 Notes Bibliogràfiques

Els conceptes introductoris de modelització conceptual que estan relacionats amb aspectes temporals es troben breument explicats en l'article [4], el qual va ser la referència inicial. En cas de desitjar informació més detallada el llibre [11] pot ser una bona referència, atès que incorpora definicions més formals mitjançant lògica de primer ordre.

La definició de restricció d'integritat i les possibles classificacions explicades són un resum d'informació obtinguda a partir del capítol 6 del llibre anteriorment indicat i del capítol 9 del llibre [10]. Pel que fa a les restriccions d'integritat temporals aplicades a les taxonomies (o més concretament a les especialitzacions) han estat obtingudes de l'article [12].

Per finalitzar, els conceptes bàsics de notació de lògica de primer ordre han estat obtinguts dels apunts de lògica de la FIB i d'algunes definicions de la wikipedia. La teoria relativa a la definició dels operadors lògics amb semàntica temporal i les característiques del temps que tracten certes lògiques temporals, es troben també explicades en l'article [3]. A nivell de suport i definicions s'ha emprat el llibre [6].

3. Restriccions d'integritat constant i permanent

3.1 Introducció

Aquest capítol defineix què són les restriccions d'integritat constant i permanent i les seves propietats. Per aquest motiu es divideix en dos parts. En la primera part es donaran les diferents variants de la definició de les restriccions constant i permanent pels tipus d'entitats, pels participants d'una relació i pels tipus de relació. Cada definició es presentarà de tres maneres: de forma textual, amb 2-FOL i amb lògica temporal. Les definicions per tipus de relacions i participants d'una relació en 2-FOL estaran formulades únicament per tipus de relacions binàries. Aquesta és una decisió per facilitar la seva comprensió. Una altre aspecte a tenir en compte és que totes les relacions d'aquest treball es consideren síncrones. Per finalitzar, en la segona part s'estudiaran les propietats de les restriccions constant i permanent.

3.2 Definició de les restriccions dinàmiques constant i permanent

3.2.1 Consideracions prèvies.

La base d'informació passa per una seqüència d'estats $\langle \sigma_0, \dots, \sigma_n \rangle$ que conforma la seva història. En aquest punt, per una correcta i més simple representació d'aquestes restriccions en lògica temporal, s'ha considerat que l'estat σ_1 és Inici(Ls) i que σ_{n-1} és Final(Ls), essent σ_0 i σ_n els estats de creació i eliminació de la base d'informació, on el sistema no existeix.

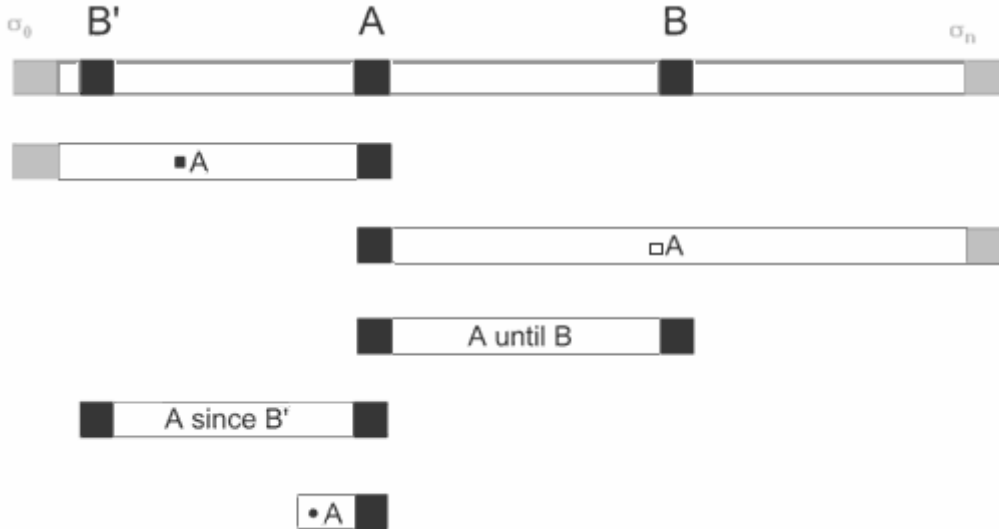


Figura 4 Semàntica dels operadors temporals

Per tant, els operadors de lògica temporal queden especialitzats de la següent manera:

La fórmula $\Box A$ és certa en l'estat σ_i si, i només si, per a tots els estats σ_j $i < j < n$ A és certa en l'estat σ_j .

La fórmula $\blacksquare A$ és certa en l'estat σ_i si, i només si, per a tots els estats σ_j $0 < j < i$ A és certa en l'estat σ_j .

La fórmula $\bullet A$ és certa en l'estat σ_i si, i només si $i > 0$ y A és certa en l'estat σ_{i-1} .

La fórmula $A \text{ until } B$ és certa en l'estat σ_i si, i només si, per algun estat σ_j , $i \leq j \leq n$, B és certa en l'estat σ_j , i per a tot k , $i < k < j$, A és certa en σ_k .

La fórmula $A \text{ since } B$ és certa en l'estat σ_i si, i només si, per algun estat σ_j , $0 \leq j < i$, B és certa en l'estat σ_j , i per a tot k , $j < k < i$, A és certa en σ_k .

3.2.2 Aplicades a un tipus d'entitat

3.2.2.1 Permanent

• Definició textual

Un tipus d'entitat E és permanent si les seves instàncies mai poden deixar de ser instància de E . Així doncs, la població d'un tipus d'entitat permanent mai pot decreïxer. Per tant, la població d'un tipus d'entitat permanent, compleix que:

1. Té un sol interval de classificació.
2. L'instant de temps final de l'interval de classificació és l'instant de temps de la fi de la vida del sistema.

Un exemple de tipus d'entitat que compleix aquesta definició seria:

- **Definició amb 2-FOL**

$$E(e,t) \rightarrow |Ci(e,E)|=1 \wedge Final(Ci(e,E)) = Final(Ls)$$

- **Definició amb lògica temporal**

$$E(e) \rightarrow \Box E(e)$$

3.2.2.2 Constant

- **Definició textual**

Un tipus d'entitat E és constant si la seva població sempre és la mateixa. És fàcil veure que els tipus d'entitat constants també són permanents. La població d'un tipus d'entitat constant, compleix que:

- Té un sol interval de classificació
- Els instants de temps inicial i final de l'interval de classificació coincideixen amb els instants de temps d'inici i fi de la vida del sistema respectivament.

Els tipus de dades (DataTypes, que en UML inclouen els enumeration) són constants per definició.

Un exemple de tipus d'entitat que compleix aquesta definició seria:

Continent

- **Definició amb 2-FOL**

$$E(e,t) \rightarrow |Ci(e,E)|=1 \wedge Final(Ci(e,E)) = Final(Ls) \wedge Inici(Ci(e,E))=Inici(Ls)$$

- **Definició amb lògica temporal**

$$E(e) \rightarrow (\blacksquare E(e) \wedge \Box E(e))$$

3.2.3 Aplicades a un participant d'un tipus de relació

3.2.3.1 Permanent

- **Definició textual**

Un tipus de relació $R(p_1:E_1, \dots, p_n:E_n)$ és permanent respecte un participant p_i , essent $1 \leq i \leq n$, si totes les instàncies de R en les que una instància de e_i de E_i participa amb el rol p_i mai deixen d'existir durant l'interval temporal en el que e_i és instància de E_i .

Un exemple seria:

HaCursat(Persona, Carrera)

Aquest tipus de relació és permanent respecte persona, atès que mentre que visqui, anirà cursant més carreres. En altres paraules, és acumulatiu. Cal ressaltar que HaCursat no té

perquè ser permanent respecte el participant carrera. Si les persones moren, desapareixen i per tant, hi ha menys persones que l'han cursat. Així doncs, la relació només seria permanent respecte carrera en cas que el tipus d'entitat Persona també ho fos.

- **Definició amb 2-FOL (suposem $i=1$)**

Amb freqüència única : $|Ci(E1, e1)|=1$

$$(R(e_1, e_2, t_1) \wedge E_1(e_1, t_2) \wedge t_2 \geq t_1) \rightarrow R(e_1, e_2, t_2)$$

Altrament

$$ic \in Ci(e_1, E_1) \wedge t_{inicial} = Inici(ic) \wedge t_{final} = Final(i_1) \wedge (t_{inicial} \leq t' \leq t'' \leq t_{final}) \wedge R(e_1, e_2, t') \rightarrow R(e_1, e_2, t'')$$

- **Definició amb lògica temporal**

$$R(e_1, \dots, e_n) \rightarrow (R(e_1, \dots, e_n) \text{ until } \neg E_i(e_i))$$

3.2.3.2 Constant

- **Definició textual**

Un tipus de relació $R(p_1:E_1, \dots, p_n:E_n)$ és constant amb respecte al participant p_i , essent $1 \leq i \leq n$, si totes les instàncies de R en les que una instància e_i de E_i participa amb el rol p_i són les mateixes durant l'interval de temps en el que e_i existeix.

Un exemple seria:

VaNéixer(Persona, País)

El tipus de relació és constant respecte persona, atès que sempre haurà nascut en la mateix país. VaNéixer respecte país no té perquè ser permanent. És anàleg a l'exemple anterior.

Un tipus de relació pot ser **derivat** i ser constant respecte un dels seus participants. En aquest cas, la regla de derivació requereix una interpretació especial: les relacions d'un participant seran sempre les que s'hagin inferit en l'instant de creació del participant.

- **Definició amb 2-FOL (suposem $i=1$)**

Si E_1 de freqüència única : $|Ci(E_1, e_1)|=1$

$$R(e_1, e_2, t_1) \wedge E_1(e_1, t_2) \wedge t_2 \geq t_1 \rightarrow R(e_1, e_2, t_2) \wedge \neg \exists e_3 (\neg R(e_1, e_3, t_1) \wedge R(e_1, e_3, t_2))$$

altrament:

$$(ic \in Ci(e_1, E) \wedge t_{inicial} = Inici(ic) \wedge t_{final} = Final(ic) \wedge R(e_1, e_2, t_{inicial})) \rightarrow (t_{inicial} \leq t \leq t_{final} \rightarrow R(e_1, e_2, t) \wedge \neg \exists e_3 (\neg R(e_1, e_3, t_{inicial}) \wedge R(e_1, e_3, t)))$$

- **Definició amb lògica temporal**

$$R(e_1, \dots, e_n) \rightarrow (R(e_1, \dots, e_n) \text{ since } \bullet E_i(e_i) \wedge (R(e_1, \dots, e_n) \text{ until } \neg E_i(e_i)))$$

3.2.4 Aplicades a un tipus de relació

3.2.4.1 Permanent

- **Definició textual:**

Un tipus de relació és permanent si ho és respecte tots els seus participants. Per tant, el conjunt d'instants finals dels intervals de classificació de les relacions de R , són subconjunt dels instants finals dels intervals de classificació de totes les entitats que participen.

Si suposem que el tipus d'entitat Persona és permanent, els tipus de relacions dels exemples anteriors serien permanents.

- **Definició amb 2-FOL:**

$$R(e_1, e_2, t) \rightarrow \text{Final}(\text{Ci}((e_1, e_2), R)) \subseteq \text{Final}(\text{Ci}(e_1, E_1)) \wedge \text{Final}(\text{Ci}((e_1, e_2), R)) \subseteq \text{Final}(\text{Ci}(e_2, E_2))$$

- **Definició amb lògica temporal:**

$$R(e_1, \dots, e_n) \rightarrow (R(e_1, \dots, e_n) \text{ until } \neg E_1(e_1)) \wedge \dots \wedge (R(e_1, \dots, e_n) \text{ until } \neg E_n(e_n))$$

3.2.4.2 Constant

- **Definició textual:**

Un tipus de relació és constant si ho és respecte tots els seus participants. Per tant, cada interval de classificació de les relacions de R té la mateixa duració que l'interval de classificació de les entitats que hi participen en R en aquell instant.

Per exemple:

$$HiPassa(Continent, Riu)$$

- **Definició amb 2-FOL:**

$$R(e_1, e_2, t) \rightarrow (\text{Ci}((e_1, e_2), R) \subseteq \text{Ci}(e_1, E_1)) \wedge (\text{Ci}((e_1, e_2), R) \subseteq \text{Ci}(e_2, E_2))$$

- **Definició amb lògica temporal:**

$$R(e_1, \dots, e_n) \rightarrow (R(e_1, \dots, e_n) \text{ since } \bullet E_1(e_1)) \wedge (R(e_1, \dots, e_n) \text{ until } \neg E_1(e_1)) \\ \wedge \dots \wedge (R(e_1, \dots, e_n) \text{ since } \bullet E_n(e_n)) \wedge (R(e_1, \dots, e_n) \text{ until } \neg E_n(e_n))$$

3.3 Les restriccions temporals constant i permanent aplicades a les reificacions

3.3.1 Introducció

La reificació d'una relació consisteix en veure-la com una entitat (és equivalent a les classes associatives d'UML). Veiem-ho amb un exemple, considerem el tipus de relació *EsMembreDe*(*membre: Persona, Comitè*). Una instància de la relació *EsMembreDe* que representa el fet que una persona és membre d'un comitè. El mateix fet podria ser vist con l'entitat, *Membre*. Quan passa això, diem que **aquesta entitat reifica la relació**.

Les instàncies de *Membre* han de ser relacionades amb les entitats de *EsMembreDe*. En cas contrari, nosaltres no coneixeríem quina relació és reificada donada una instància de *Membre*. La correspondència entre les instàncies d'un *Membre* i aquelles de *EsMembreDe* són establertes pel significat de un tipus de relació binària per cada participant de *EsMembreDe*. En l'exemple tenim dos tipus de relacions binàries.

$R1(\text{Membre}, \text{Persona})$

$R2(\text{Membre}, \text{Comitè})$

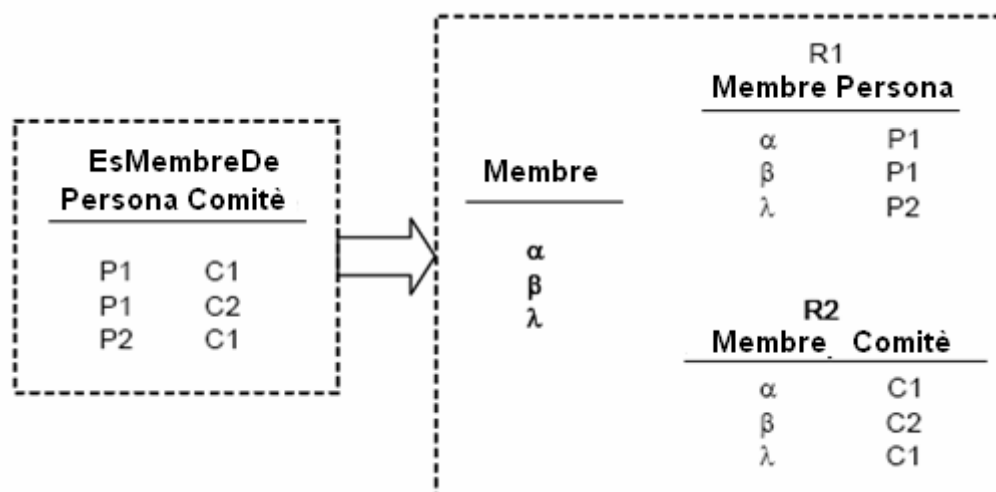


Figura 5 Exemple d'instanciació d'una reificació

Aquests tipus de relacions són anomenades intrínseques degut a que poden ser vistes com a part del nou tipus d'entitat.

Formalment, la reificació d'un tipus de relació $R(p_1:E_1, \dots, p_n:E_n)$ és un tipus d'entitat E i n tipus de relacions intrínseques $R(p:E; p_i:E_i)$, ($i=1..n$). S'han de complir les següents propietats:

- Les R_i són constants respecte p . Per tat una instància de E sempre es refereix a la mateixa relació de R , i que $\text{Card}(p;p_i;R_i) = (1,1)$. Llavors, hi ha una dependència funcional $\{p\} \rightarrow \{p_i\}$ en R_i .
- $\text{Card}(p_i;p; R_i) = \text{Card}(p_i; p_1, \dots, p_{i-1}, p_{i+1}, \dots, p_n; R)$
- Hi ha una correspondència un a un entre les poblacions de R i E :

$$R(e_1, \dots, e_n) \rightarrow \exists e (E(e) \wedge R_1(e, e_1) \wedge \dots \wedge R_n(e, e_n))$$

$$R_1(e, e_1) \wedge \dots \wedge R_n(e, e_n) \rightarrow R(e_1, \dots, e_n)$$

Aquesta definició garanteix que puguem en qualsevol moment obtenir les instàncies de R a partir de les instàncies de E i les instàncies dels seus tipus de relacions intrínseques.

3.3.2 Semàntica de les restriccions constant i permanent aplicades a una reificació

Com hem vist una reificació correspon a que un tipus d'entitat sigui vist com un tipus de relació i al revés. Per tant, és interessant veure quina definició de les restriccions constant i permanent s'aplica en aquests casos.

Primerament veurem que significat té que el tipus d'entitat que reifica a algun tipus de relació sigui constant o permanent. Seguidament veurem quin significat té que el tipus de relació que es reificada sigui constant o permanent.

3.3.2.1 Semàntica de les restriccions constant i permanent en un tipus d'entitat que reifica a un tipus de relació

Si és **permanent** un tipus d'entitat E que reifica a un tipus de relació $R(p_1:E_1, \dots, p_n:E_n)$, tal i com s'ha definit formalment en el punt 3.2.2.1 per tipus d'entitats permanents, significa que les seves instàncies mai poden deixar de pertànyer a E . Per tant, atès que hi ha una correspondència un a un entre les poblacions de R i E , les instàncies de la relació que representa mai es podran eliminar durant la vida del sistema.

Seguint l'exemple anterior, si el tipus d'entitat *Membre* fos permanent, significaria que les relacions *EsMembreDe* mai podrien eliminar-se de la base d'informació. És a dir que un cop que una persona esdevé membre d'un comitè, ho serà fins el final de la vida del sistema. Per tant, el tipus de relació reificada és permanent.

En el cas que sigui **constant** un tipus d'entitat E que reifica a un tipus de relació $R(p_1:E_1, \dots, p_n:E_n)$, significa, tal i com s'ha definit formalment en el punt 3.2.2.2 per a tipus d'entitats constants, que la seva població és sempre la mateixa durant la vida del sistema. Per tant, atès que hi ha una correspondència un a un entre les poblacions de R i E , les instàncies de la relació que representa mai es podran ni crear ni eliminar des de l'inici fins a la fi del sistema.

Sigui el següent exemple: en el domini d'una escola d'idiomes, el tipus d'entitat *Curs* reifica el tipus de relació *EsCursa*(*Idioma*, *Nivell*). En aquesta escola d'idiomes, sempre s'ensenyen els mateixos cursos. Conseqüentment *Curs* és constant. Això comporta que les relacions de *EsCursa* sempre siguin les mateixes des de l'inici a la fi del sistema. Per tant, aquest tipus de relació és constant.

3.3.2.2 Semàntica de les restriccions constant i permanent en tipus de relacions reificades

Si un tipus de relació $R(p_1:E_1, \dots, p_n:E_n)$ és **permanent** i reificada per un tipus d'entitat E , significa que és permanent respecte tots els seus participants. És important veure que això no implica que necessàriament E sigui permanent. Ho ensenyarem amb el següent exemple: en el domini d'una empresa on s'enregistren els responsables que ha tingut un departament i que són actualment membres de l'empresa. Els actuals membres de l'empresa són representats pel tipus d'entitat *Empleat*. També es considera

en aquest domini que *Responsable* és un tipus d'entitat que reifica el tipus de relació *esResponsableDe(Empleat, Departament, Data)*. Aquest tipus de relació es permanent respecte tots els seus participants, però *Responsable* no. El motiu és que un empleat pot deixar de treballar per l'empresa, i per tant les instàncies de *Responsable* d'aquest empleat serien eliminades de la base d'informació en el mateix moment en el que s'eliminés la instància que representa aquest empleat.

Si un tipus de relació $R(p_1:E_1, \dots, p_n:E_n)$ és **constant** reificada per un tipus d'entitat E , significa que és constant respecte tots els seus participants. És important veure que això no vol dir que necessàriament E sigui constant o permanent. Es pot veure amb el següent exemple: suposem que un sistema emmagatzema les variants d'un idioma. Una variant d'un idioma és d'una zona geogràfica. Per exemple, el català té com a variants el lleidatà, gironès, barcelonès, Per tant podem considerar que el tipus de relació *VariaEn(Idioma, ZonaGeogràfica)* pot estar reificada pel tipus d'entitat *VariantLingüística*.

El tipus de relació és constant respecte els dos participants, però les instàncies es poden haver insertat durant la vida del sistema, per exemple quan es dona d'alta un idioma. Per tant, *VariantLingüística* no té perquè ser constant.

3.4 Propietats relatives a l'esquema estructural de les restriccions temporals constant i permanent

Hi trobem tres conjunts de propietats:

- **Relatives als tipus de relacions:** estudiarem si partir d'un tipus de relació que és constant o permanent respecte a algun dels seus participants, infereix que altres elements d'un esquema conceptual també compleixin aquest tipus de restriccions.
- **Relatives a reificacions:** estudiarem quines implicacions tenen els participants d'un tipus de relació reificada en un tipus d'entitat i al invers.
- **Relatives a especialitzacions:** estudiarem quines implicacions té, donada una especialització/generalització, el fet de que el supertipus o el subtipus sigui constant o permanent.

3.4.1 Relatives als tipus de relacions

Totes les demostracions per tipus de relacions, estan formulades per tipus de relacions binàries, però es fàcil comprovar que es poden estendre a tipus de relacions n-àries.

Partirem de la següent afirmació:

Si un tipus de relació $R(p_1:E_1, p_2:E_2)$ és permanent respecte el participant p_1 , la participació del participant p_2 pot ser no permanent.

Serà demostrat per reducció al absurd. Suposem la següent hipòtesi:

Si la relació és permanent respecte el participant p_1 , la participació del participant p_2 és permanent.

i que tenim la següent instanciació de la relació R :

p_1	p_2
α_1	β_1
α_1	β_2

Taula 2 Instanciació exemple

Atès que R és síncrona, al finalitzar un interval temporal en el que $E_1(\alpha_1)$ és cert, també deixarà de ser cert que $R(\alpha_1, \beta_1)$ i $R(\alpha_1, \beta_2)$. Si encara es satisfà $E_2(\beta_1)$ o $E_2(\beta_2)$ i R és permanent respecte p_2 , arribem a una contradicció: existeix un interval temporal en que el nombre d'instàncies de la relació R ha decrescut i encara és cert $E_2(\beta_1)$ o $E_2(\beta_2)$.

Exemple:

*HaVisitat(Persona, País),
on Persona no té població permanent i País té població constant.*

Atès que constant encara és més restrictiu que permanent, també hem demostrat que:

- Si un tipus de relació $R(p_1:E_1, p_2:E_2)$ és permanent respecte el participant p_1 , la participació del participant p_2 pot ser no permanent.
- Si un tipus de relació $R(p_1:E_1, p_2:E_2)$ és constant respecte el participant p_1 , la participació del participant p_2 pot ser no permanent
- Si un tipus de relació $R(p_1:E_1, p_2:E_2)$ és constant respecte el participant p_1 , la participació del participant p_2 pot ser no constant.

Propietat 1

Si un tipus d'entitat E_1 és permanent i el tipus de relació $R(p_1:E_1, p_2:E_2)$ és permanent respecte el participant p_1 , el tipus de relació és permanent respecte el participant p_2

Serà demostrat per reducció al absurd. Suposem la hipòtesi inversa:

Si un tipus d'entitat E_1 és permanent i el tipus de relació $R(p_1:E_1, p_2:E_2)$ és permanent respecte el participant p_1 , la relació pot ser no permanent respecte el participant p_2

Per tant, el nombre d'instàncies del tipus de relació R en les que una instància de E_2 hi participa com a p_2 pot decreïxer en l'interval temporal en el que existeix com a E_2 .

Donada la següent instanciació:

p_1	p_2
α_1	β_1
α_2	β_1

Taula 3 Instanciació propietat 1

Tenim que $E_1(\alpha_1)$ i $E_1(\alpha_2)$ sempre serà cert, atès que la població de E_1 és permanent. Si a més, R és permanent respecte p_1 , mai desapareixeran les instàncies de R . D'altra banda, si R respecte p_2 no és permanent, per definició pot succeir que hi hagi instàncies de R que deixin d'existir. Per tant, hem arribat a una contradicció.

Exemple:

*HaVisitat(Persona,País),
on Persona té població permanent i País té població constant.*

*Coneix(subjecte:Persona, conegut: Persona)
on Persona té població permanent i Coneix és un tipus de relació recursiva*

Amb la segona propietat i atès que constant és més restrictiu que permanent, també demostrem de forma equivalent que:

- Si un tipus d'entitat E_1 és permanent i el tipus de relació $R(p_1:E_1, p_2:E_2)$ és constant respecte el participant p_1 , la relació és permanent respecte el participant p_2 .
- Si un tipus d'entitat E_1 és constant i el tipus de relació $R(p_1:E_1, p_2:E_2)$ és constant respecte el participant p_1 , la relació és permanent respecte el participant p_2 .

Propietat 2

Si el tipus entitat E_1 és constant i el tipus de relació $R(p_1:E_1, p_2:E_2)$ és constant respecte el participant p_1 , la relació és constant respecte el participant p_2 .

Demostració de que R respecte p_2 ha de ser permanent: per la demostració anterior tenim que és cert per que el fet que E_1 sigui constant i que la seva participació sigui constant és un cas particular de que siguin permanents. Per tant, la demostració és com la de la propietat anterior.

Demostració de que no pot ser que R respecte p_2 sigui permanent i no sigui constant: ho demostrarem per reducció a l'absurd. Suposem que R respecte p_2 no és constant. Llavors, es podria afegir a la instanciació de la Taula 2, una relació $R(\alpha_3, \beta_1)$ en qualsevol instant del Ls. No obstant, si la participació R respecte p_1 és constant, i E_1 és constant, $R(\alpha_3, \beta_1)$ hauria d'haver existit durant tot el Ls. Per tant, no és cert que s'hagués pogut afegir un cop començat el Ls. Conseqüentment, s'ha arribat a una contradicció.

Exemple:

VaHabitat (Continent, Dinsaure)
En un tipus de relació recursiva:
EsAnteriorAlfabèticament(primera:Lletra, segona:Lletra)

Propietat 3

Siguin dos tipus d'entitats E_1 i E_2 , on E_1 és permanent. Sigui $R(p_1:E_1, p_2:E_2)$ una relació síncrona on:

- **La participació de E_1 és considerada permanent.**
- **$\text{CardMin}(E_2;E_1;R)>0$ (participació obligatòria)**

Llavors E_2 també és permanent.

Es demostrarà per reducció a l'absurd partint de la següent instanciació:

T a	E_1	$R(p_1:E_1, p_2:E_2)$	E_2
	e_1	$R(e_1, e_2)$	e_2

Taula 4 Instanciació propietat 3

Suposem que E_2 no és permanent:

Llavors podem eliminar una entitat $e_2 \in E_2$. Recordem també que totes les entitats de E_2 participen en alguna relació de R perquè la $\text{CardMin}(E_2;E_1;R)>0$. Per tant, si eliminem una entitat e_2 , al ser síncrona R , també eliminarem totes les relacions en les que participava e_2 , que com a mínim hi haurà una.

La participació de E_1 en R és permanent, cosa que significa que mentre una entitat e_1 existeixi, no poden decreixer el nombre de relacions en les que participa. En aquest cas, atès que E_1 és permanent, mai podrem eliminar cap relació. Hem arribat a una inconsistència, i per tant, a una contradicció.

Exemple:

HaTreballat(Empleat, Departament),
essent Empleat un tipus d'entitat permanent.

Propietat 4

Siguin dos tipus d'entitats E_1 i E_2 , on E_1 és constant. Sigui $R(p_1:E_1, p_2:E_2)$ una relació síncrona on:

- **La participació de E_1 és considerada constant.**
- **$\text{CardMin}(E_2; E_1; R) > 0$ (participació obligatòria)**

Llavors E_2 també és constant.

Es demostrarà per reducció a l'absurd. Si E_1 és constant i la seva participació en R també, aquestes instàncies de la relació estaran durant tot el Ls. A més, per la propietat 2 sabem que la participació de E_2 també és constant. Ara falta demostrar que el tipus d'entitat E_2 també és constant. Suposem que E_2 no és constant:

Llavors pot haver un instant de temps t_1 , tal que $t_f > t_1 > t_0$, (essent t_0 i t_f els instants d'inici i fi del Ls respectivament), en el que es pot crear una entitat de tipus E_2 . Al crear-la, sabent que la $\text{CardMin}(E_2; E_1; R) > 0$, implica que hem de instanciar com a mínim una relació de R en t_1 . Com que $t_0 < t_1$ i prèviament hem anunciat que totes les relacions de $R(p_1:E_1, p_2:E_2)$ estaran durant tot el Ls, no podem instanciar una relació de R en t_1 , ni tampoc podem crear una entitat de tipus E_2 . Per tant, hem arribat a una contradicció.

Exemple:

HiPassa(Continent, Riu)

3.4.2 Relatives a reificacions

Propietat 5

Si algun dels participants d'un tipus de relació reificada és un tipus d'entitat permanent i la seva participació és permanent, llavors el tipus d'entitat E resultant de la reificació és permanent.

Ho demostrarem per reducció a l'absurd. Suposem que E no és permanent, llavors poden eliminar-se instàncies d'aquest tipus. Si eliminéssim una entitat $e \in E$, seria equivalent a eliminar la relació que representa. Això comportaria també eliminar les entitats que tenen participació permanent. Atès que per l'enunciat, com a mínim una d'elles correspon a un tipus d'entitat permanent, no ho podríem fer perquè violaríem dita restricció. Per tant, arribem a una contradicció.

Exemple:

En el domini d'un sistema de reserves de cotxes, tenim que el tipus d'entitat *Reserva* reifica el tipus de relació *Reserva(Usuari, Model, Data)*. En aquest sistema és necessari tenir constància de totes les reserves fetes d'una data (o sigui qualsevol). Això significa que la participació de *Data* és permanent. Atès que *Data* és un *DataType*, és un tipus d'entitat constant, i per tant permanent, no es podran eliminar les reserves del sistema, i per tant,

Reserva és permanent. Remarquem que *Usuari* no té perquè ser permanent, ja que si no ha fet cap reserva, es pot eliminar del sistema. El mateix passaria per Model.

Propietat 6

Si algun dels participants d'un tipus de relació reificada és un tipus d'entitat constant i la seva participació és constant, llavors el tipus d'entitat E resultant de la reificació és constant.

Per la propietat anterior, i atès que constant encara és més restrictiu que permanent, tenim que E és permanent. Per tant només queda per demostrar que E no pot ser que sigui permanent i no constant. També ho farem per reducció a l'absurd.

Suposem que E fos un tipus d'entitat permanent però no fos constant, llavors podríem afegir una entitat $e \in E$ durant la vida del sistema. Si això succeís, com que E reifica un tipus de relació, equivalentment estariem creant una relació nova entre diverses entitats. D'acord amb l'enunciat, com a mínim una de les participacions és constant. Per tant, per fer possible la inserció en la base d'informació d'aquesta nova relació complint les restriccions de participació constant, hauríem de crear les corresponents instàncies d'entitats que tenen dita participació. Recordem, que també com a mínim una d'elles pertany a un tipus d'entitat constant, i causaria que violéssim aquesta restricció. Per tant, hem arribat a una contradicció.

Exemple:

Tornant a l'exemple de les variants lingüístiques del punt 3.3.2.2, on *VariantLingüística* era un tipus d'entitat que reifica al tipus de relació *VariaEn(Idioma, ZonaGeogràfica)*. Suposem que tenim un sistema on la participació de *ZonaGeogràfica* és constant i la seva població també. Llavors totes les instàncies de *VariantLingüística* haurien de ser les mateixes durant la vida del sistema per no violar aquestes dues restriccions. Per tant, *VariantLingüística* ha de ser constant.

Propietat 7

Si un tipus d'entitat que reifica a un tipus de relació $R(p_1:E_1, \dots, p_n:E_n)$ és permanent, llavors tots els participants de R tenen participació permanent¹.

Per la propietat de la reificació que afirma que hi ha una correspondència un a un entre les poblacions de R i E:

$$\begin{aligned} R(e_1, \dots, e_n) &\rightarrow \exists e(E(e) \wedge R_1(e, e_1) \wedge \dots \wedge R_n(e, e_n)) \\ R_1(e, e_1) \wedge \dots \wedge R_n(e, e_n) &\rightarrow R(e_1, \dots, e_n) \end{aligned}$$

¹ Pot ser una combinació de participacions constant i permanent.

tenim que les instàncies de R romandran en la base d'informació fins al final de la vida del sistema. Això ve a dir que totes les seves participacions hagin de ser permanents.

Exemple:

En el domini d'un sistema de reserves de cotxes, on el tipus d'entitat *Reserva* reifica el tipus de relació *Reserva(Usuari, Model, Data)*. Un dels requisits d'aquest sistema es tenir constància de totes les reserves que s'han fet. Per tant, el tipus d'entitat *Reserva* és permanent. Això implica que un usuari cada estat de la base d'informació haurà realitzat més reserves, un model serà cada cop més demandat i en una data s'hauran fet més reserves. Per tant, com podem veure, totes les participacions són permanents.

Propietat 8

Si un tipus d'entitat E que reifica a un tipus de relació $R(p_1:E_1, \dots, p_n:E_n)$ és permanent, llavors totes les E_i , $i=(1, \dots, n)$ amb participació obligatòria són permanents.

Si E és permanent, per la propietat anterior R és permanent. Per tant hem de garantir que cap instància de la seva població s'elimini de la base d'informació. Quelcom podria pensar que això implica que tots els tipus d'entitats que hi participen han de ser permanents. Això no és del tot cert perquè estaríem afegint la restricció a que qualsevol instància d'un d'aquests tipus, encara que no hi participi no pogués ser esborrat del sistema. D'altra banda també es podria pensar que només és necessari que alguna ho sigui. Tampoc és cert, atès que estaríem també afegint la mateixa restricció innecessària del cas anterior, però només per un tipus d'entitat. En canvi, si una d'aquestes entitats participa obligatòriament en la relació, sí que hem de restringir que aquesta sigui permanent. El motiu és el següent: qualsevol instància d'aquesta entitat que participa com a mínim en una relació de R , com que cap relació d'aquest tipus pot ser eliminada, tampoc pot ser eliminada cap entitat d'aquest tipus.

Exemple:

Tornem a emprar el domini d'un sistema de reserves de cotxes, on tenim que el tipus d'entitat *Reserva* reifica el tipus de relació *Reserva(Usuari, Model, Data)*. En aquest sistema segueix sent necessari tenir constància de totes les reserves fetes, i a més que només s'enregistrin els usuaris que han fet alguna reserva. Veiem que la participació d'usuari esdevé obligatòria, i per tant *Usuari* ha de ser un tipus d'entitat Permanent.

Propietat 9

Si un tipus d'entitat que reifica a un tipus de relació $R(p_1:E_1, \dots, p_n:E_n)$ és constant, llavors tots els participants de R tenen participació constant

Per la propietat de la reificació que afirma que hi ha una correspondència un a un entre les poblacions de R i E :

$$R(e_1, \dots, e_n) \rightarrow \exists e (E(e) \wedge R_1(e, e_1) \wedge \dots \wedge R_n(e, e_n))$$

$$R_1(e, e_1) \wedge \dots \wedge R_n(e, e_n) \rightarrow R(e_1, \dots, e_n)$$

tenim que les instàncies de R sempre seran les mateixes. Per tant, totes les seves participacions han de ser constant.

Exemple:

Tornant a l'exemple de les variants lingüístiques del punt 3.3.2.2, on *VariantLingüística* era un tipus d'entitat que reifica al tipus de relació *VariaEn(Idioma, ZonaGeogràfica)*. Suposem que *VariantLingüística* és constant. Això implicaria que un Idioma sempre es parlés sempre en les mateixes zones geogràfiques, i que en una zona geogràfica sempre es parlés el mateix idioma.

Propietat 10

Si un tipus d'entitat E que reifica a un tipus de relació $R(p_1:E_1, \dots, p_n:E_n)$ és constant, llavors totes les E_i , $i=(1, \dots, n)$ amb participació obligatòria són constants.

Si E és constant, per la propietat anterior R és constant. Per tant hem de garantir que cap instància de la seva població s'elimini de la base d'informació. Quelcom podria pensar que això implica que tots els tipus d'entitats que hi participen han de ser constants. Això no és del tot cert perquè estaríem afegint la restricció a que qualsevol instància d'un d'aquests tipus, encara que no hi participi no pogués ser creada ni esborrada del sistema. D'altra banda també es podria pensar que només és necessari que alguna ho sigui. Tampoc és cert, atès que estaríem també afegint la mateixa restricció innecessària del cas anterior, però només per un tipus d'entitat. En canvi, si una d'aquestes entitats participa obligatòriament en la relació, sí que hem de restringir que aquesta sigui constant. El motiu és el següent: qualsevol instància d'aquesta entitat que participa com a mínim en una relació de R, com que cap relació d'aquest tipus pot ser creada ni eliminada durant la vida del sistema, tampoc pot ser creada ni eliminada cap entitat d'aquest tipus.

Exemple:

Tornem a l'exemple de les variants lingüístiques del punt 3.3.2.2, on *VariantLingüística* era un tipus d'entitat que reifica al tipus de relació *VariaEn(Idioma, ZonaGeogràfica)*. Suposem que *VariantLingüística* és constant i que la participació d'*Idioma* és obligatòria. Llavors, el tipus d'entitat *Idioma* és constant.

3.4.3 Relatives a especialitzacions

Les demostracions de les següents propietats 11 i 12 són anàlogues per especialitzacions de tipus de relacions.

Una especialització, que no compleixi alguna de les condicions donades per les següents propietats, no s'ha trobat que es pugui inferir si els subtipus són constants o permanents. Per les propietats 5, 6 i 7 cal a dir que els subtipus o supertipus corresponen a **només un nivell d'especialització**.

Propietat 11

Una especialització que compleix:

- És absoluta o relativament estàtica
- El supertipus té població permanent

qualsevol subtipus té població permanent.

Si l'especialització és absoluta o relativament estàtica, qualsevol instància del supertipus *E*, si aquesta ho és d'algun dels subtipus, sempre ho serà mentre sigui de *E*. Atès que *E* és permanent, implica pel fet anterior, que qualsevol subtipus té població permanent.

Exemple:

Sigui *Empleat* un tipus d'entitat permanent que té com a subtipus d'una especialització *Programador*, *Arquitecte* i *Analista* sota la dimensió *perfil professional*. En l'empresa d'aquest exemple un empleat no pot canviar de perfil (especialització absolutament estàtica). Per tant, si s'entrés un nou programador seria una nova instància sempre d'*Empleat* i de *Programador* i mai deixaria de ser-ho durant la vida del sistema.

Propietat 12

Una especialització que compleix:

- És absolutament estàtica
- El supertipus té població constant

qualsevol subtipus té població constant.

Per la demostració anterior, tenim que tots els subtipus tenen població permanent. Només queda demostrar que tenen població constant.

Si un subtipus no tingués població constant, podria donar-se el cas que és donés d'alta una nova instància durant la vida del sistema. Si passes estem davant dos casos:

- Aquesta instància ja estigui prèviament classificada en el supertipus: En aquest cas, arribem a una contradicció o estat no permès pel fet que és una especialització absolutament estàtica.

- Donar d'alta la instància en el supertipus i en el subtipus: En aquest cas arribem a una contradicció o estat no permès atès que estaríem donant d'alta una instància d'una entitat amb població constant.

Exemple:

Podem emprar el mateix que el de la propietat anterior però essent el tipus d'entitat Empleat constant en comptes de permanent.

Propietat 13

Una partició que compleix que:

- És relativament dinàmica
- El supertipus té població constant

Qualsevol subtipus té població constant

Per la demostració anterior, tenim que tots els subtipus tenen població permanent. Només queda demostrar que tenen població constant.

Si un subtipus no tingués població constant, podria donar-se el cas que és donés d'alta una nova instància durant la vida del sistema. Si passés estem davant dos casos:

- Aquesta instància ja estigui prèviament classificada en el supertipus: Arribaríem a una contradicció o estat no permès per les restriccions de *disjointness*, *covering* i relativament dinàmica.
- Donar d'alta la instància en el supertipus i en el subtipus: En aquest cas arribem a una contradicció o estat no permès atès que estaríem donant d'alta una instància d'una entitat amb població constant.

Exemple:

Sigui el supertipus el tipus d'entitat constant Instruments. Aquest té com a subtipus d'una partició relativament dinàmica (és menys restrictiva que totalment estàtica): Corda, Percussió, Vent i Electrònic. Per tant, si donéssim d'alta en el sistema una flauta, aquesta seria una instància d'instrument de Vent i conseqüentment d'instrument. Atès que Instrument té població constant, hem arribat a una contradicció o estat no permès. Per tant, la flauta hauria d'haver estat introduïda a l'inici de la vida del sistema.

Propietat 14

Una taxonomia que :

- Té tots els subtipus finals amb població permanent.
- En el possible diagrama d'estats, totes les transicions porten com a estat final a algun dels subtipus finals.

Llavors el supertipus té població permanent.

Ho demostrarem per reducció a l'absurd. Suposem que el supertipus no té població permanent. Si no té població permanent, podem eliminar instàncies del supertipus. Si les eliminem mai arribaran al subtipus final, i per tant serà inconsistent amb les transicions definides en el diagrama de transició, perquè com hem dit, sempre porten a algun subtipus finals. Hem arribat a una contradicció.

Exemple:

En el context d'una empresa de lloguers de cotxes:

Venut isA Cotxe

EnLloguer isA Cotxe

EnReparació isA Cotxe

Disponible isA Cotxe

On Venut seria l'estat final.

Per tant, Cotxe ha de ser permanent.

3.5 Propietats relatives a l'esquema de comportament de les restriccions temporals constant i permanent

3.5.1 Conceptes previs

Un event estructural és un canvi elemental en la població d'un tipus d'entitat o de relació. En aquest punt s'estudien les condicions que hi ha d'haver entre els events estructurals i un esquema estructural on apareixen elements que han de complir les restriccions de constant i permanent.

Un event de domini, és un conjunt d'un o més events estructurals que passen en un instant determinat. Per tant, defineix els canvis permesos entre dos estats consecutius de la base d'informació. És important comentar que en un event de domini només s'indiquen aquells events estructurals no derivats, és a dir aquells que no són conseqüència d'altres per mantenir la base d'informació consistent. En aquest capítol s'ha considerant que els events de domini són modelats com a tipus d'entitats.

Quelcom podria pensar que atès que en un event de domini s'especifiquen quins canvis són permesos, veritablement s'han d'estudiar les relacions entre els events de domini i les restriccions de constant i permanent en comptes dels events estructurals. El motiu principal pel qual aquesta afirmació no és certa és el següent: els events estructurals poden esborrar o afegir elements de la base d'informació, però sempre la instància creada es diferent de l'eliminada. És a dir, pot ser que el valor dels atributs i relacions siguin els mateixos, però té diferent oid, i per tant es considera una instància diferent.

3.5.2 Propietats

Propietat 15

No pot haver un event de estructural que creï o elimini instàncies d'un tipus d'entitat E constant.

Si un tipus d'entitat és constant, no es poden ni afegir ni esborrar instàncies de E. Per tant, cap canvi d'estat pot incloure instàncies d'un tipus d'event estructural corresponent a les insercions i esborrats de E.

Propietat 16

Sigui la relació $R(p_1:E_1, \dots, p_n:E_n)$ i $1 \leq i \leq n$. Si E_i té una participació constant en R, llavors només es poden crear relacions de R quan un event estructural crea una entitat de E_i .

Utilitzem el següent contraexemple per validar la propietat: Suposem que existeixen dos tipus d'entitats anomenades *Empleat* i *Departament* i un tipus de relació *Treballa*(*Empleat*, *Departament*). També suposem que un empleat sempre treballa al mateix departament. Per tant, empleat té participació constant en aquest tipus de relació. Seguint aquest domini, si tenim la següent instanciació:

Empleat(Joan)
Departament(TIC)
Departament(RRHH)
Treballa(Joan, TIC)

Si hi hagués un canvi d'estat que instàncies un event estructural en el que crees la relació *Treballa*(Joan, RRHH), estariem violant la restricció de constant. Per tant dit event estructural no pot existir.

Propietat 17

No pot haver un event de estructural que elimini instàncies d'un tipus d'entitat E permanent.

Si un tipus d'entitat és permanent, no es poden esborrar instàncies de E. Per tant, cap canvi d'estat pot incloure instàncies d'un tipus d'event estructural corresponent als esborrats de E.

Propietat 18

Sigui un tipus de relació $R(p_1:E_1, \dots, p_n:E_n)$ i $1 \leq i \leq n$. Si E_i té una participació permanent en R, llavors només es poden eliminar les relacions de R en que participa una entitat $e_i \in E_i$ quan s'elimina e_i .

Utilitzem el següent contraexemple per validar la propietat: Suposem que

existeixen dos tipus d'entitats anomenades *Persona* i *Empresa* i un tipus de relació *HaTreballat(Persona, Empresa)*. Aquesta relació indica les empreses en les que ha treballat una persona al llarg de la seva vida. Per tant, persona té participació permanent en aquest tipus de relació. Seguint aquest domini, si tenim la següent instanciació:

Empleat(Joan)
Empresa(BBVA)
Empresa(LaCaixa)
HaTreballat(Joan, BBVA)
HaTreballat(Joan, LaCaixa)

Si hi hagués un event de domini que crees la instància d'un event estructural en el que s'eliminés la relació *HaTreballat(Joan, BBVA)*, i no creés la instància d'un event estructural que eliminés *Empleat(Joan)* estariem violant la restricció de permanent, atès que estem perdent una de les empreses en les que ha treballat *Joan*. Per tant aquesta situació no s'ha de provocar.

Propietat 19

Sigui una relació $R(p_1:E_1, \dots, p_n:E_n)$, $1 \leq i \leq n$, $1 \leq j \leq n$ amb $i \neq j$. Si E_j amb té una participació permanent en R , llavors si eliminem una entitat $e_i \in E_i$, tal que existeix una relació $R(e_i, \dots, e_j)$, també s'eliminarà l'entitat e_j .

Tornem a l'exemple de la propietat anterior i suposem que s'instancia en un instant t un event de domini. Segons l'especificació d'aquest event de domini, s'instancia un d'estructural que elimina de la base d'informació l'entitat *Empresa(LaCaixa)*. Com el tipus de relació és síncrona (tal i com s'indica a la introducció d'aquest capítol), també s'ha d'instanciar un altre event estructural que elimini de la base d'informació *HaTreballat (Joan, LaCaixa)*. El fet que es produeixi aquest event estructural, per tal de satisfer la propietat anterior també s'haurà instanciar un altre que elimini *Empleat(Joan)*.

Propietat 20

No pot haver un event de estructural que modifiqui el valor d'un atribut en una entitat ja existent, si aquest és constant.

Un atribut equival a una relació. Per tant, si la participació de l'entitat que conté l'atribut en aquesta relació és constant, per definició de participació constant, el valor de l'atribut no es pot modificar, ja que això equivaldria en esborrar i crear una nova relació del mateix tipus i participant.

3.6 Notes bibliogràfiques

Atès el bàsic coneixement dels elements relatius de l'esquema de comportament i sobre les reificacions, aquests no s'han explicat en el capítol de l'estat de l'art. La referència bibliogràfica empleada és el capítol 11 de [10].

4. Importància pràctica de les restriccions constant i permanent.

4.1 Introducció

Un llenguatge de modelització conceptual permet mitjançant símbols, representar gràficament tipus d'entitat i de relació. Disposar d'una simbologia per expressar restriccions d'integritat aporta, en cas de que aquestes siguin molt freqüents, una major agilitat en el desenvolupament i manteniment d'esquemes conceptuais. Un clar exemple d'aquesta realitat succeeix amb les restriccions de cardinalitat.

Conseqüentment, s'ha considerat interessant investigar el percentatge d'aparició de les restriccions constant i permanent en l'esquema estructural de tres casos d'estudi. Possiblement, aquests casos d'estudi tinguin certes similituds amb altres sistemes d'informació del món de la indústria. Mitjançant aquesta tasca podrem saber si seria convenient estendre un llenguatge de modelització conceptual per poder-les representar.

Aquest capítol es divideix en tres parts. En la primera part es presenten els casos d'estudi de eBay[14] , EU-Rent Car Rentals[8] i DBLP[13]. En la segona part es mostra quin percentatge d'aparició tenen les restriccions constant i permanent en els casos d'estudi. En la tercera i última part explica les conclusions dels resultats obtinguts.

4.2 Casos d'estudi

Els diferents casos d'estudi corresponen a tres treballs realitzats en el departament. A continuació, per cada cas d'estudi s'ha fet un resum dels aspectes que tracten, i així comprendre millor els conceptes que apareixen en els respectius annexos.

4.2.1 Reenginyeria d'un sistema de mercat electrònic: eBay

eBay és una empresa coneguda internacionalment on es realitzen subhastes. La gent pot comprar i vendre mercaderia i serveis per arreu del món a través seu. La seva especificació divideix el model conceptual de e-Bay amb cinc sub-esquemes conceptuais:

- **Usuaris, articles i categories**

En e-Bay es distingeixen tres tipus d'usuaris: visitants, usuaris i venedors. Un visitant només pot accedir a informació sobre els articles oferts al mercat electrònic.

Un usuari és aquell que realitza compres i totes les seves dades romandran per sempre al sistema. Per a que un visitant esdevingui usuari d'eBay ha d'omplir un formulari. Seguidament rebrà un correu amb el seu codi per realitzar les compres.

Els venedors són usuaris que fan ofertes d'articles. Aquests poden contractar diferents extres a eBay per poder accedir a una venda més ràpida.

Els articles són aquells productes que es poden comprar en eBay. Els articles estan classificats en categories per facilitar-ne la cerca per part dels usuaris o visitants de la pàgina.

- **Oferta d'articles**

Els usuaris que s'hagin enregistrarat com a venedors podran oferir els seus articles per a que la resta d'usuaris puguin mostrar el seu interès per adquirir-los. En aquest punt s'explica les diferents maneres d'oferir articles que proporciona eBay. Principalment hi ha dues maneres: articles posats en venda, és a dir, articles oferts amb la intenció (i obligació per part dels interessats) de completar la transacció econòmica, i els articles només anunciats, oferts amb la intenció de contactar amb els possibles compradors. Si es vol completar la transacció econòmica (no és un bé immoble) hi ha diverses modalitats: la venda immediata, subhasta estàndard i subhasta múltiple. En canvi si només es vol contactar per compradors, (es tracta d'un bé immoble) només es permet la publicació d'un anunci.

- **Oferta i demanda**

La demanda apareix quan un usuari està interessat a comprar un article ofert. Si aquest usuari resulta ser un dels usuaris que més diners està disposat a pagar per l'article ofert, serà un dels guanyadors de l'oferta, tenint l'obligació d'acabar realitzant la compra (sempre i quan no es tracti d'un immoble). En el cas de que no sigui un immoble, l'usuari guanyador no tindrà l'obligació d'acabar comprant l'article, sinó que guanyarà la possibilitat de posar-se en contacte amb el venedor si l'anunci és en format subhasta. Entenem per guanyador aquell usuari o usuaris que estan disposats a pagar una major quantitat monetària per una

determinada oferta, ja sigui una oferta de venda o una d'anunci en subhasta. Les ofertes en format d'anunci simple no tenen guanyadors, ja que tots els demandants podran entrar en contacte amb el venedor. Per tant el tipus de demanda variarà en funció del tipus d'oferta associada a l'article. Totes les vendes quedaran enregistrades per sempre en el sistema.

- **Després de la venda: Comentaris.**

eBay és un lloc web que permet l'adquisició d'articles oferts pels usuaris. El fet que els articles oferts pertanyin a particulars comporta un cert risc pels compradors, ja que tot el que poden saber sobre l'article que els interessa és allò que en diu el venedor. Per tant, pot no ser objectiu.

Una solució a aquest problema és deixar constància del seu grau de satisfacció en la realització de la compra. Així, quan acabi l'oferta i aquesta consti com acceptada pel venedor, tant aquest com el comprador o compradors poden deixar un comentari per a l'altra part acompanyat d'una valoració positiva, negativa o neutra que reflecteixi la seva satisfacció sobre el comportament de l'altre usuari envers la translació de l'article en qüestió.

- **Taxes**

S'expliquen les taxes imposades per eBay per cadascun dels serveis oferts. Existeixen les següents taxes imposades :sobre el fet d'oferir un article, sobre el possible preu mínim establert quan es tracta d'una subhasta estàndard, sobre l'import final de l'article quan aquest acabi venent-se i sobre els extres que el venedor hagi decidit incorporar a l'oferta. Per tant, totes les taxes que es cobren a eBay recauen sobre l'oferent, mai sobre el demandant.

4.2.2 EU-Rent Car Rentals Specification

Correspon a l'especificació d'una empresa que es dedica al lloguer de cotxes. Té sucursals en diferents països. Els seus serveis inclouen entre d'altres, la reserva anticipada de cotxes i diferents formes de pagament. En aquest cas, la modelització dels casos d'ús està dividida en *packages*, corresponents a les diferents àrees temàtiques, per donar una major facilitat de comprensió. Hi ha les següents:

- **Gestió de reserves**

Hi ha dos tipus de reserves. Les que es s'accepten en l'acte (walk-in rental) i les es fan per anticipat, les quals es confirma la seva realització en funció de les condicions ofertes per l'empresa. Tota reserva passa per diferents estats en funció del tipus, però mai desapareix del sistema, arribant a un dels següents estats: cancel·lada pel client, cancel·lada per la companyia i tancada per devolució del cotxe.

- **Gestió de clients**

Aquest sistema enregistra tots els clients que ha tingut. El motiu principal és que els interessa tenir constància de l'història d'activitats que ha comès qualsevol client per saber si ha estat problemàtic en reserves anteriors. Si no ho ha estat, se li donarà la possibilitat d'oferir-li els seus serveis un cop més. En cas contrari, el client s'afegirà en la llista negra. També hi ha un tipus de clients privilegiats, els que pertanyen al grup *loyalty incentive scheme*. Un client passa a pertànyer a aquest grup quan ha fet quatre reserves en un any. En aquest moment podrà pagar les reserves amb els punts que acumuli d'altres reserves.

▪ Preus i Descomptes

Aquesta empresa sempre intenta que el preu que hagi de pagar el client sigui mínim. Per tant, s'escollirà el preu mínim entre l'instant de realització de la reserva (aplicant els descomptes d'aquell moment) o l'instant en el que estigui marcat com a data de devolució de cotxe (també aplicant els corresponents descomptes). Això obliga també a mantenir un històric de preus. A més, el fet que no desapareguin les reserves del sistema, obliga a mantenir els preus anteriors, atès que sinó, el preu d'una reserva feta seria incorrecte.

▪ Assignació de cotxes

Un cotxe d'un model o grup en concret sol·licitat en una reserva pot no estar disponible en la sucursal de recollida del cotxe. Darrere hi ha tota una gestió per transportar el conjunt necessari de cotxes en entre sucursals pel dia següent. Això comporta que un cert cotxe, de la mateixa manera que les reserves, passi per un conjunt d'estats. S'explica amb més detall en "Gestió de cotxes".

▪ Preparació i manteniment de cotxes

Quan un cotxe és retornat a una sucursal, si fa tres mesos des de l'última revisió de manteniment o bé ha acumulat 10000 km passa a entrar en revisió. En la revisió es comprova si el cotxe supera els 4000 km. per posar-ho a venda o no. També pot ser que el cotxe tingui danys o bé que estigui en perfectes condicions. En el primer cas, s'haurà de reparar i s'enregistra com una mala experiència del client. En el segon cas, el cotxe passarà a estar disponible.

▪ Recollida i devolució de cotxes

La recollida es pot fer o bé quan es fa una reserva immediata, o bé quan passa el dia fixat de la reserva. En cas de que en tot el dia el cotxe no hagi estat recollit, passa a estar disponible per a una altra reserva.

La reserva és pagada en el moment en el que el cotxe és retornat, encara que pot ser retornat amb retràs, en una sucursal no acordada o bé abans d'hora. Si és tornat amb retràs el seu preu serà incrementat i s'enregistrarà com una mala experiència del client. Si és tornat en una sucursal no acordada, també s'afegirà un suplement al preu. I, si és tornat abans d'hora, es calcularà el nou preu d'acord als dies que ha durat el lloguer.

▪ Gestió de cotxes

L'empresa gestiona quins cotxes són ja vells, i per tant ha de vendre, i quants cotxes i de quin model ha de comprar de nous. Ja s'ha comentat que un cotxe passa per un conjunt d'estats: en transferència: disponible, assignat, usat, en checkeig, en reparació, en manteniment, en venda, i venut. Mai hi ha un estat en que s'elimini un cotxe del sistema, sempre estarà en un estat, essent sempre l'estat final "venut".

▪ Gestió de sucursals

EU-Rent Car Rentals ofereix els seus serveis en diferents sucursals que estan distribuïdes geogràficament en diferents països. Cada sucursal pertany a un tipus de sucursal.

Al fer una reserva, és obligatori enregistrar les sucursals de recollida i de retorn del cotxe. Per tant, per a que la informació de la reserva sigui coherent, mai s'han de poder eliminar les sucursals que han existit en el sistema que han fet alguna reserva (que per normalitat, s'han considerat totes).

▪ Grups de cotxes i gestió de models

Un cotxe és d'un model concret, i els models es classifiquen en grups. Un grup es crea per classificar cotxes amb unes característiques similars. Cada grup varia en funció de la sucursal i té una quota, la qual correspon al nombre desitjable de cotxes d'un tipus en concret. No s'assumeix que hi ha un nombre fixat de grups de cotxes.

▪ Gestió i millora d'indicadors

Cada tipus de sucursal té uns propis indicadors per poder fer càlculs heurístics. En funció dels valors que donin aquests heurístics el responsable de la sucursal realitzarà una acció o una altra. Es defineix com acció qualsevol funcionalitat que permeti el sistema.

4.2.3 DBLP

DBLP, una pàgina web amb bibliografia de ciències de computació, va ser originalment una base de dades i un lloc amb continguts bibliogràfics de programació lògica.

El cas d'estudi relacionat només conté parts de l'esquema estructural del sistema DBLP: persones (que poden ser autors o editors) i les seves publicacions que són llibres editats o publicacions (llibres, capítols i papers de revistes). Els capítols i els papers de revistes poden o no ser llibres de conferències.

4.3 Notació i Resultats

A continuació es mostren els resultats obtinguts dels annexos. Es calculen en cada cas d'estudi pels següents conjunts d'elements²:

- Tipus d'entitats (T. d'entitat excloent DataTypes)
- Tipus de relació (T. de relació)
- Participacions (Participacions)
- Atributs.

Total : El seu valor correspon al total de cada conjunt d'elements de cada cas d'estudi.

Permanent: El seu valor correspon al total, en cada grup d'elements, que compleixen la restricció de permanent. El número d'atributs permanents és el número d'entitats (o reificacions) que tenen una participació permanent en dit atribut.

Constant : El seu valor correspon al total, en cada grup d'elements, que compleixen la restricció de constant. El número d'atributs constant és el número d'entitats (o reificacions) que tenen una participació constant en dit atribut.

Percentatge: El seu valor correspon al percentatge d'elements de cada conjunt que compleix les restriccions de constant i permanent.

² Les reificacions s'han comptat tant com a tipus de relació que com a tipus d'entitat, ja que la semàntica de constant i permanent varia, tal i com s'explica al capítol 3.

eBay	Total	Permanent	Constant	%
T. de entitat	54	42	2	81
T. de relació	34	12	0	35
Participacions	77	36	5	53
Atributs	109	0	58	53

Taula 1. Resultats del cas d'estudi eBay

EU-Rent Car Rentals	Total	Permanent	Constant	%
T. de entitat	56	31	3	61
T. de relació	59	25	0	42
Participacions	119	29	18	39
Atributs	69	0	31	44

Taula 2 Resultats del cas d'estudi EU-Rent Car Rentals

DBLP	Total	Permanent	Constant	%
T. de entitat	17	15	0	88
T. de relació	18	7	11	100
Participacions	36	7	29	100
Atributs	38	0	30	78

Taula 3 Resultats del cas d'estudi DBLP

4.4 Conclusions

El percentatge d'aparició de les restriccions constant i permanent és molt elevat en els dos casos d'estudi. Supera al 30% en tots els casos. Pot donar-se el risc que s'ha donat la particularitat que aquests dos casos d'estudi siguin dels pocs en que s'aconsegueixin aquestes xifres. De totes maneres, s'han escollits exemples reals, i possiblement aquests tinguin moltes similituds amb altres sistemes del món de la indústria. Per consegüent, el risc és menor.

Si un tipus de restricció apareix habitualment, possiblement sigui convenient poder-la representar gràficament en un llenguatge de modelització conceptual. Així aconseguim agilitzar la tasca de desenvolupament i manteniment de l'analista. La representació proposada es troba en el capítol anterior.

4.5 Notes Bibliogràfiques

L'única referència necessària en aquest capítol correspon als tres casos d'estudi tractats [14,13,8].

5 Representació proposada en UML de les restriccions constant i permanent

5.1 Introducció

El *Llenguatge Unificat de Modelatge (UML)* és un llenguatge per especificar, visualitzar, construir i documentar els artefactes dels sistemes software. UML ofereix un estàndard per a construir esquemes d'un sistema, incloent-hi aspectes conceptuals com processos de negoci i funcions del sistema, i d'altres aspectes més concrets, ja siguin esquemes de bases de dades o components software reusables.

L'UML va ser adoptat com a estàndard de l'*Object Management Group, Inc. (OMG)* l'any 1997. L'OMG és una organització internacional formada per: venedors de sistemes d'informació, desenvolupadors de software i usuaris. Va ser fundada al 1989, i promociona tant la teoria com la pràctica de la tecnologia d'orientació a objectes pel desenvolupament de software. Les funcions de l'organització inclouen l'establiment de pautes per a la indústria i especificacions per a la gestió d'objectes per establir un marc comú pel desenvolupament d'aplicacions. Els objectius principals són la reusabilitat, portabilitat i interoperabilitat del software basat en objectes d'entorns heterogenis.

Per aquest motiu, s'ha considerat important realitzar un capítol que examina com es poden representar les restriccions temporals: constant i permanent.

Aquest capítol es divideix en quatre parts. La primera part és una breu introducció al metaesquema d' UML. La segona part y la tercera part són les explicacions de com es tracten els aspectes de canviabilitat, en la versió 1.4 y 2.2 d'UML respectivament. La quarta part completa el metaesquema actual d'UML per representar gràficament les restriccions temporals de constant i permanent tal i com ja s'explica en [10].

d'entitat o dataType i tenen una multiplicitat. Per aquest motiu, hi ha una relació entre Property i Classifier (*type*). Una instància d'un tipus de relació en UML rep el nom de **link**.

La multiplicitat no està reflectida en aquest fragment d'esquema, però a la versió completa apareix un tipus d'entitat abstracta, anomenada **MultiplicityElement**, que és supertipus de Property. MultiplicityElement conté, entre d'altres atributs, la cardinalitat mínima i màxima acceptada.

El tipus d'entitat **Generalization** té com a instàncies els tipus de relacions d'especialitzacions (**isA**). Una especialització té un supertipus, modelat amb el rol **general**, i un subtipus, modelat amb el rol **specific**.

En condicions estrictes, UML emplea rectangles per les classes. Els tipus d'entitats haurien de ser definits com a classes amb l'estereotip «type». Una classe estereotipada «type» especifica un domini d'objectes sense definir una implementació física d'aquests objectes. Per tant, per una millor lectura quan parlem de tipus d'entitats, si no es diu el contrari, es farà referència a les classes d'UML amb l'estereotip type. D'altra banda, en les gràfics en UML posteriors, s'assumirà en cas de que no hi hagi cap altre estereotip paraula clau que estem modelant tipus d'entitats.

5.3 Representació de les restriccions temporals constant i permanent en UML 1.4

La propietat més lligada a les restriccions temporals constant i permanent és la **canviabilitat**. Aquesta propietat només afecta als atributs i als participants d'un tipus de relació. Aquest punt estudiarà el significat dels possibles valors d'aquesta propietat en la versió d'UML 1.4 i estudiarà si es podrien modelitzar les restriccions de constant i permanent en el cas de participacions d'un tipus de relació, i conseqüentment de relacions amb aquests valors.

5.3.1 Conceptes previs

Presentem algunes definicions necessàries particulars del metaesquema d'UML 1.4:

StructuralFeature: tipus d'entitat abstracta, que té només com a subtipus el tipus d'entitat Attribute.

Attribute: tipus d'entitat que té com a instàncies els atributs d'un esquema conceptual en UML.

AssociationEnd: tipus d'entitat que té com a instàncies les participacions d'un tipus de relació.

5.3.2 Fragments del metaesquema

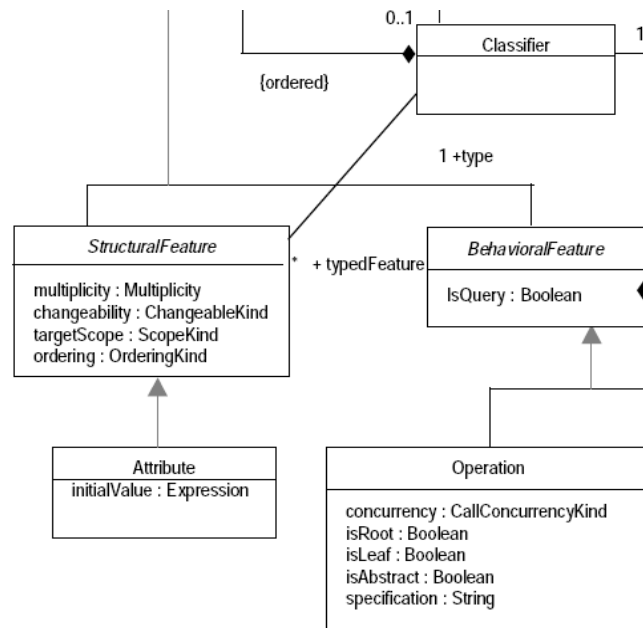


Figura 7 Fragment del metaesquema de la versió 1.4 d'UML associat a la canviabilitat dels atributs

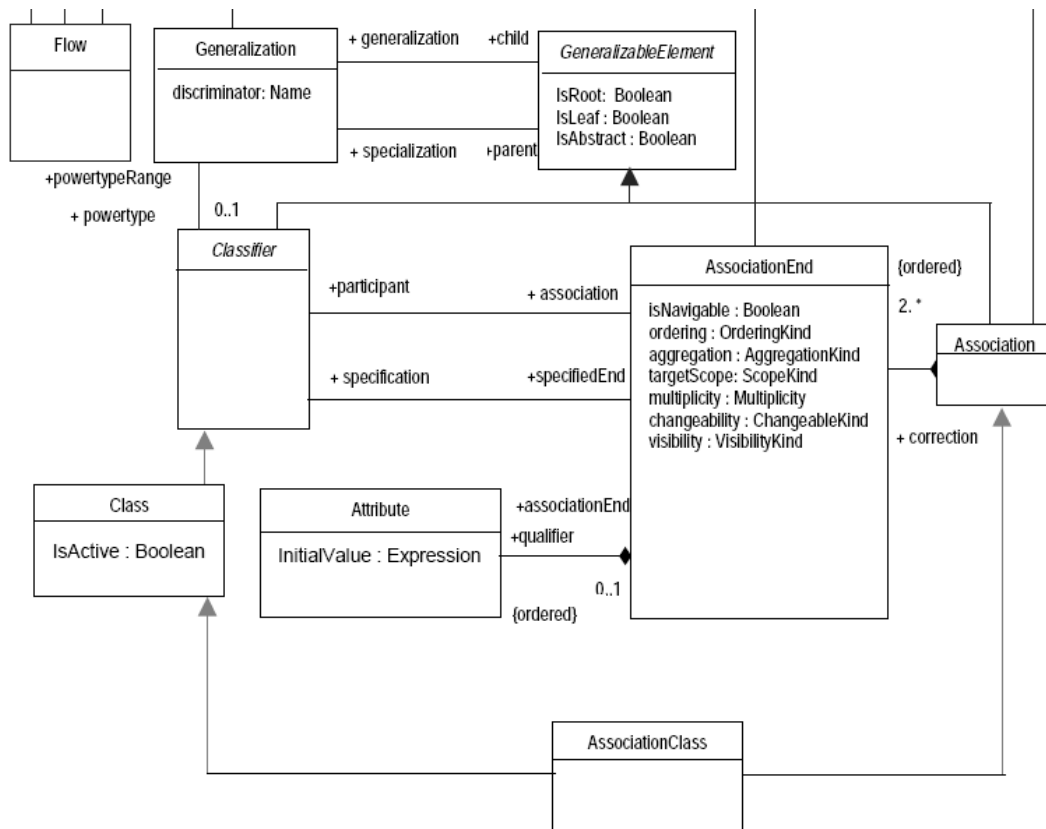


Figura 8 Fragment del metaesquema de la versió 1.4 d'UML associat a la canviabilitat dels AssociationEnd

5.3.3 La canviabilitat

El metaesquema inclou una enumeració anomenada ***ChangeableKind***. Representa els valors que pot prendre la canviabilitat en atributs i extrems d'una associació. A continuació presentem les tres definicions existents per cada tipus de valor de canviabilitat: per **ChangeableKind**, per **Attribute** i per **AssociationEnd**.

ChangeableKind

- **changeable**: No hi ha restriccions de modificació.
- **frozen**: El valor no pot canviar des de l'extrem l'origen de la relació després de la creació i inicialització de l'objecte origen. Les operacions de l'altre extrem sí que poden canviar el valor.
- **addOnly**: si la multiplicitat no és fixada, poden ser afegits valors en qualsevol moment des de l'objecte origen, però un cop creat un valor, no pot ser eliminat des de l'extrem l'origen. Les operacions de l'extrem destí poden canviar el valor.

Attribute

- **changeable**: No hi ha restriccions de modificació.
- **frozen**: El valor de l'atribut no pot ser alterat després de que l'objecte és instanciat i els seus valors inicialitzats. No es poden afegir valors a un conjunt.
- **addOnly**: Té sentit si la multiplicitat no és fixada en un únic valor. Valors addicionals poden ser afegits al conjunt de valors, però una vegada creat un valor, aquest no pot ser ni eliminat ni modificat.

AssociationEnd:

- **changeable**: No hi ha restriccions de modificació.
- **frozen**: no es poden afegir links per operacions de la classe origen després de la creació de l'objecte origen. Les operacions de la classe destí poden afegir links (probablement no estigui igualment de restringit).
- **addOnly**: Els links poden afegir-se en qualsevol moment per operacions de l'objecte origen, però un cop creat un link, no pot ser eliminat per operacions de la classe origen. Les operacions de la classe destí poden afegir o esborrar relacions (probablement no estigui igualment de restringit).

5.3.4 Notació

Si els links són canviabls (poden ser afegits, eliminats, i modificats) no és necessari cap indicador. La propietat {frozen} indica que no es poden afegir, esborrar o moure links des d'un objecte (que té com a extrem oposat d'una associació aquest indicador) després que l'objecte és creat i inicialitzat. La propietat {addOnly} indica que es poden afegir links (presumiblement, si la multiplicitat és variable); encara que els links no poden ser modificats ni esborrats.

Per tant, un exemple de la seva representació seria:

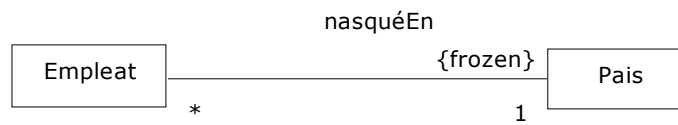


Figura 9 Exemple en UML d'una participació frozen en una associació

o bé, pel cas dels atributs:

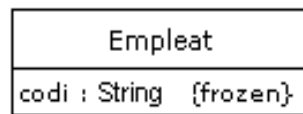


Figura 10 Exemple en UM d'una participació frozen en un atribut

5.3.5 Equivalència amb les restriccions temporals constant i permanent

Intuïtivament sembla que frozen sigui equivalent a una participació constant i addOnly a una participació permanent, però no és cert. Els valors de canviabilitat en UML són restriccions sobre el comportament de les operacions d'una classe que té navegabilitat cap a un extrem marcat com {addOnly} o {frozen}. Això comporta que, per una banda, l'operació *effect()* d'un domaint event les pugui violar. D'altra banda, en modelització conceptual la navegabilitat és en els dos sentits, i constant i permanent són restriccions que s'han de respectar de forma global, no només des de les operacions d'una classe. Això comporta que l'ús de les restriccions frozen i addOnly com a notació de les restriccions constant i permanent, puguin portar a que aquestes no es respectin si no hi ha navegabilitat.

Suposem una relació binària $R(p_1:E_1;p_2:E_2)$ on:

- La participació de p_1 és constant. Llavors, la marquem com a frozen en l'extrem p_2 .
- En el disseny, només fa falta navegabilitat de E_2 cap a E_1 .

La navegabilitat indicada permet que la classe E_2 pugui afegir o eliminar instàncies del tipus de relació R . Per tant, sempre es seguirà respectant el valor de canviabilitat frozen, atès que des de la classe E_1 cap operació canvia els links de R . Però el més important és donar-se compte que al poder la classe E_2 eliminar o afegir links de R des de les seves operacions, pot ser que no es respecti la restricció d'integritat constant del participant p_1 .

A més, s'ha vist que atribut/relació derivat, que és constant o permanent, és caracteritzat perquè sempre tindrà el mateix valor. Això és degut a que la regla de derivació tindrà implícit que ha de ser el valor calculat en la seva inicialització. La versió 1.4 d'UML, no indica la semàntica dels valors de canviabilitat per atributs o relacions derivades, per tant, aquest queda indeterminat.

5.4 Representació de les restriccions temporals constant i permanent en UML 2.1.1

La canviabilitat en la versió 2.1.1 d'UML és tracta diferent que en les versions anteriors. A continuació estudiem els mateixos punts que a l'apartat anterior però amb la nova versió.

5.4.1 Conceptes previs

StructuralFeature: tipus d'entitat abstracta que conté tots els elements de tipus estructural que no són ni tipus d'entitat ni tipus de relacions representats en un esquema UML. Només té com a subtipus Property.

Property: tipus d'entitat que té com a instàncies els atributs i les participacions d'un tipus de relació d'un esquema conceptual en UML.

5.4.2 Fragments del metaesquema

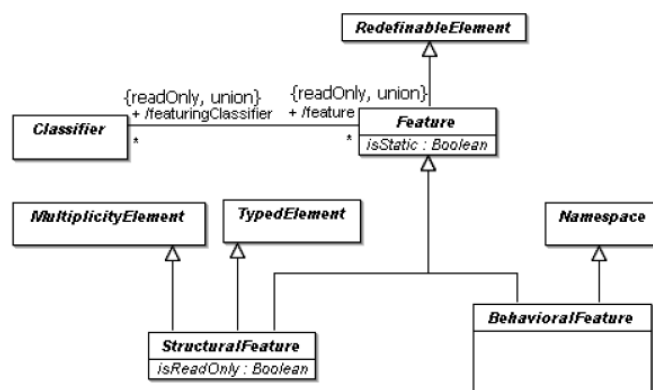


Figura 11 Fragment del metaesquema de la versió 2.1 d'UML associat a la canviabilitat

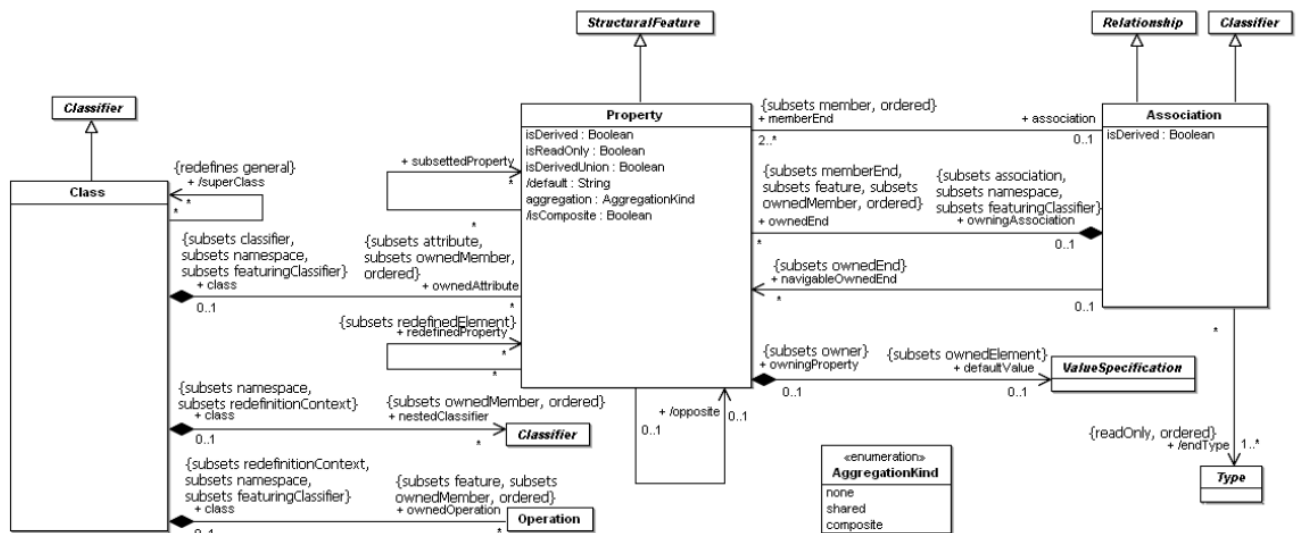


Figura 12 Fragment del metaesquema de la versió 2.1 d'UML associat a la canviabilitat d'una Property

5.4.3 La canviabilitat

La canviabilitat és una propietat representada amb un atribut de Property anomenat *isReadOnly*. Aquest atribut és de tipus booleà. A continuació veiem la seva definició del manual de referència.

isReadOnly: si és cert, l'atribut només pot ser llegit, i no escrit. El valor per defecte és fals.

L'atribut *isReadOnly* està relacionat amb dos restriccions d'integritat del metaesquema:

- la propietat de *isNavigable* pot ser marcada com *isReadOnly*.

isReadOnly **implies** *isNavigable()*

- si una property té *isDerivedUnion* a cert, llavors *isReadOnly* és cert.

isDerivedUnion **implies** *isReadOnly*

Aquest concepte està imprecisament definit. Clarament, una property ha de ser modificable inicialment mentre és inicialitzat, però el concepte d'inicialització no està definit en UML. Probablement un rang de valors, com en UML1 però amb més opcions, seria millor que un simple valor booleà.[15]

També s'indica en [15] que si una property està marcada com a derivada aquesta molt freqüentment és *readOnly*; si és canviable, la semàntica d'actualització del valor estarà específicament implementada.

5.4.4 Notació

Hi ha dos paraules reservades per la canviabilitat: {readOnly} i {unrestricted}. La paraula {readOnly} és posat en la property si aquesta no pot ser modificada. En cas contrari, pot indicar-se amb {unrestricted}. Generalment s'assumeix que l'absència de {readOnly} implica que la property pot ser modificada. Exemples:

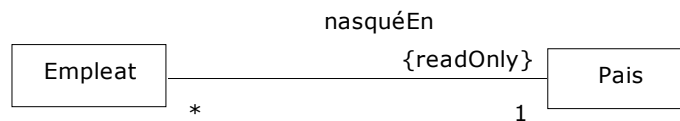


Figura 13 Exemple d'ús de readOnly en associacions

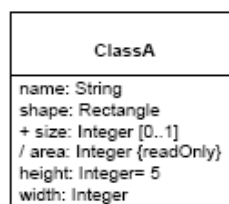


Figura 14 Exemple d'ús de readOnly en atributs

5.4.5 Equivalència amb les restriccions temporals constant i permanent

Podríem arribar a pensar que és equivalent representar mitjançant isReadOnly les restriccions d'integritat constant d'un participant respecte un tipus de relació, i conseqüentment, els tipus de relació constant. No és del tot cert. Per una part, que un atribut només es pugui llegir no implica que aquest no pugui canviar, ja que aquest pot ser derivat. A més, si un atribut és derivat per unió, no implica que aquest sigui constant, tal i com afirma la segona restricció. Per tant, no podem representar les restriccions constants mitjançant {readOnly}.

5.5 Representació proposada de les restriccions temporals constant i permanent en UML

5.5.1 Representació d'entitats amb població constant o permanent

UML utilitza un rectangle amb un nom per representar entitats. Com ja hem dit, en UML no es pot formalitzar definir que un tipus d'entitat és constant o permanent. Nosaltres proposem assumir que hi ha dos restriccions predefinides (constant i permanent) que la seva formalització és implícita: les eines que analitzen els esquemes o generen el codi entenen el seu significat i actuaran en conseqüència.

En UML hi ha un mecanisme anomenat *stereotype* per definir noves restriccions predefinides.

Un stereotype és una classe en la que les seves instàncies extenen les característiques definides en un element del model. Aquestes noves característiques són informació addicional que pot ser usada per diversos propòsits.

Els stereotypes són definits amb el símbol de classe (el rectangle) amb la paraula reservada «stereotype». Per tant d'aquesta manera indiquem nous tipus d'entitat del metaesquema.

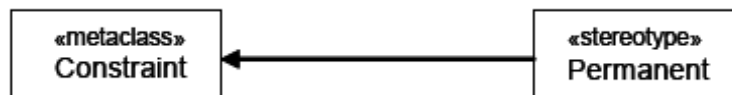


Figura 15 Definició l'estereotip permanent

Pel moment, sembla suficient que tipus d'entitat constant o permanent escrivint { «constant» } o { «permanent» }, respectivament, prop del nom del tipus d'entitat. El text entre claus { C } situat prop del símbol del tipus d'entitat significa que la població del tipus d'entitat ha de satisfer la restricció C. En aquest cas, la restricció és el stereotype «constant» o «permanent», el qual significa que són restriccions predefinides. Un exemple:



Figura 16 Exemples de l'ús dels estereotips constant i permanent en tipus d'entitat

5.5.2 Representació de participacions constants o permanents.

Donada una relació $R(p_1:E_1, p_2:E_2, \dots, p_n:E_n)$ si la participació de E_1 és constant o permanent, es posa { «constant» } o permanent { «permanent» } en l'extrem de la línia de la relació R que toca a E_1 al costat de p_1 en cas de que estigui escrit.

Un exemple:

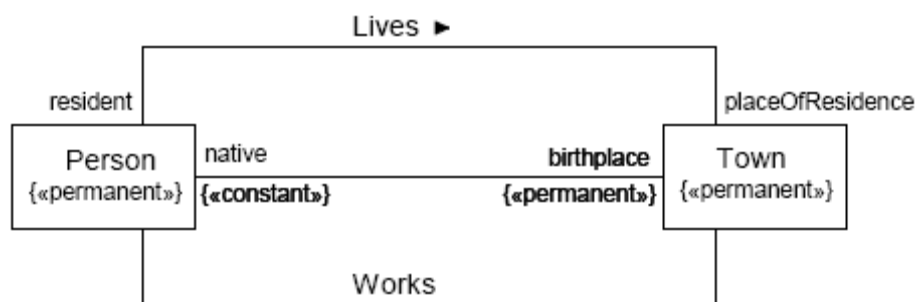


Figura 17 Exemple de l'ús dels estereotips constant i permanent

En aquest cas, la participació de native seria constant i la participació de birthplace seria permanent. Pels atributs s'usaria la mateixa simbologia al costat dret on apareix el tipus de l'atribut.

5.6 Notes bibliogràfiques

Per realitzar aquest capítol s'han necessitat informació sobre l'especificació dels metamodels de UML 1.4 [1] i 2.1[2] i consultar els comentaris de [15]. D'altra banda em justificat la representació de les restriccions constant i permanent que proposava[10]

6 Conclusions

El resultat d'aquesta tesi ha estat un estudi de les restriccions temporals constant i permanent. Primerament, hem contextualitzat aquestes restriccions i explicat la lògica temporal i de primer ordre. Seguidament, empleant conceptes temporals de modelització conceptual, hem proposat una definició formal mitjançant aquestes dues lògiques i un conjunt de vint propietats. Totes aquestes propietats han estat demostrades formalment. Poder validar els models que tenen les restriccions constant i permanent amb les propietats definides equivaldria a validar un subconjunt de requisits i notar en alguns casos l'absència d'alguns d'ells per a que l'especificació fos consistent. Cal recordar que això suposa un avenç perquè estem detectant errors en l'especificació que d'altra manera podrien ser propagats en etapes posteriors en el desenvolupament de sistemes d'informació i per tant un cost de temps i esforços addicionals.

Mitjançant l'anàlisi dels esquemes conceptuals de tres casos d'estudi reals, hem arribat a la conclusió que aquestes restriccions apareixen amb molta freqüència (més del 30%). Aquest fet justifica la conveniència de disposar d'una notació gràfica específica per definir-les, tal i com passa amb les restriccions de cardinalitat. D'aquesta manera s'agilitza el procés d'especificació, atès que no s'haurien de definir textualment. Ademès, aquest percentatge representa aproximadament l'impacte de millora que suposa el nostre treball segons el que s'ha comentat en el punt anterior.

S'ha examinat si el metamodel del nucli de les versions 2.1 i 1.4 d'UML proporcionava algun tipus d'element per definir-les, i s'ha trobat que la notació empleada per aspectes de canviabilitat no era l'ajustada en cap dels casos. Per tant, era necessari ampliar el nucli mitjançant els estereotips constant i permanent.

El treball descrit pot continuar en varies direccions. En primer lloc, pot ser interessant modelar les restriccions constant i permanent amb l'extensió OCL proposada en [4]. En segon lloc, seria convenient analitzar altres esquemes conceptuals de sistemes reals per corroborar o no que les restriccions estudiades mereixen un tractament particular. En tercer lloc, es pensa implementar aquestes restriccions en un entorn de modelització concret, incloent en control de les propietats trobades. Per finalitzar, seria interessant proposar patrons de disseny senzills per verificar en temps d'execució que es satisfan aquestes restriccions.

7 Bibliografia

1. Unified Modeling Language Specification. Version 1.4.2. *Final Adopted Specification formal/04-07-02*.
2. Unified Modeling Language: Superstructure. Version 2.1.1. *Final Adopted Specification formal/2007-02-03*. 2.
3. BELLINI, P., R. MATTOLINI, and P. NESI. 2000. Temporal Logics for Real-Time System Specification. *ACM Computing Surveys*. 32(1),.
4. CABOT, J., A. OLIVÉ, and E. TENIENTE. 2003. Representing Temporal Information in UML. *The Unified Modeling Language: Model Languages and Applications, proceedings of 6th International Conference, October*. pp 44-59.
5. CHEN, P. 1976. The Entity-Relationship Model: Towards a Unified View of Data. *ACM Transactions on Database Systems*. 36(9),.
6. CHOMICKI, J. and G. SAAKE. 1998. *Logics for Databases and Information Systems*. Anon. Kluwer Academic Publishers Boston.
7. COSTAL, D., A. OLIVÉ, and M. R. SANCHO. 1997. Temporal Features of Class Populations and Attributes in Conceptual Models. *16th International Conference on conceptual Modelling (ER'97), Los Angeles, November*. pp 57-70.
8. FRIAS, L., A. OLIVÉ, and A. QUERALT. 2003. *EU-Rent Car Rentals Specification*. [EU-Rent Car Rentals Specification] LSI-03-59-R. AnonUPC: .
9. HALPIN, Terry. 2001. *Information Modeling and Relational Databases. from Conceptual Analysis to Logical Design*. Anon. San Francisco: Morgan Kaufmann Publishers. ISSN 1400429565.
10. OLIVÉ, A. 2007. *Conceptual Modeling of Information Systems*. Anon. Springer Verlag.

11. OLIVÉ, A. 2001. *Modelització Conceptual De Sistemes d'informació*. Anon. Edicions UPC.
12. OLIVÉ, A., D. COSTAL, and M. R. SANCHO. 1999. Entity Evolution in ISA Hierarchies. *Proceedings of the 18th International Conference on Conceptual Modeling*. pp 62-80.
13. PLANAS, E. and A. OLIVÉ. 2006. The DBLP Case Study.
14. QUERALT, A. Reenginyeria d'Un Sistema De Mercat Electrònic.
15. RUMBAUGH, James,. *The Unified Modeling Language Reference Manual Second Edition*. Anon. Boston: Addison-Wesley. ISSN 0-321-24562-8.
16. VAN GRIETHUYSEN, JJ. *Concepts and Terminology for the Conceptual Schema and the Information Base*. AnonISO/TC97/SC5, 1982.

Annexos

7.1 Cas d'estudi *EU-Rentcar*

BRANCH

Tipus d'entitats

Nom	Tipus	Motiu
BrachType	Constant	S'assumeix que estan totes creades i que hi ha una per cada nom de tipus de sucursal.
Country	Permanent	<p>Els països enregistrats seran aquells pels quals es passarà amb cotxes llogats o bé on hi hagi una branch.</p> <p>Amb els motius següents :</p> <ul style="list-style-type: none"> Els tipus d'entitats amb els que manté alguna relació són de població permanent (Branch, BranchType,...) o pot ser un subtipus(GuaranteedReservation, OpenedRental, etc..). Les participacions de Country en les diferents relacions és permanent. No apareix un event que esborri entitats d'aquest tipus. <p>sembla natural considerar que té una naturalesa permanent, encara que fet certes verificacions podria no exigir-se.</p> <p>-A remarcar que a l'esquema estructural la cardinalitat Card(Country; BranchType;CountryBranchType) és errònia i hauria de ser (0,infinít), degut a que hi ha countries que només serveixen per registrar els països pels quals passarà un cotxe d'una reserva.</p> <p>DE: RecordCountryInformation,</p>
PerformanceIndicator	Variable	<ul style="list-style-type: none"> No hi ha cap DE que elimini entitats d'aquest tipus El sistema seguiria complint els requisits del sistema si la població d'aquest tipus d'entitat és variable. <p>DE: CreateIndicator.</p>
Branch	Permanent	<ul style="list-style-type: none"> Si es poguessin esborrar entitats d'aquest tipus, s'hauria de comprovar que no hi hagués cap relació amb alguna reserva. El motiu és que es vol tenir un històric, i per tant, produiria un esborrat de reserves, encara que no ho hagués sol·licitat el client o l'empresa (Reservation no té població permanent), i això no és admissible. <p>Com a conseqüència, s'ha suposat que la població és permanent per simplicitat, però sembla poc provable que hi hagi una branch on no s'hagi fet cap lloguer.</p> <p>DE: CreateBranch</p>
Car	Permanent	<ul style="list-style-type: none"> Tots els estats pels que un car deixi d'usar-se de lloguer en EU-Rent Car són representats amb diferents tipus d'entitats. Per tant és una mica il·lògic que a part hi hagi un DE per eliminar entitats de Car del IB. <p>DE:ReceiveCar</p>
OwnCar	Permanent	<p>Els mateixos que Car, no hi ha diferència entre Car i OwnCar.</p> <p>DE: ReceiveCar</p>
PendantCarOrder	Variable	<ul style="list-style-type: none"> No hi ha cap DE que elimini entitats d'aquest tipus, però tampoc és presenten problemes si es permetés esborrar entitats d'aquest tipus. Sembla coherent que quan arribi el cotxe nou sol·licitat, la comanda desaparegui del sistema, en el mateix moment o bé quan hagi passat la garantia del cotxe, si es que és vol conservar per aquest motiu. <p>DE:OrderCar</p>
CarModel	Permanent	<ul style="list-style-type: none"> Car és permanent Card(CarGroup;Car; /isGroup)=(1, infinit) i per inferència Card(CarModel;Car)=(1, infinit) Car és constant respecte CarModel en la relació isOfA <p>DE: CreateCarModel</p>

Nom	Tipus	Motiu
BrachType	Constant	S'assumeix que estan totes creades i que hi ha una per cada nom de tipus de sucursal.
CarGroup	Permanent.	<ul style="list-style-type: none"> El rentGroup, hauria de ser constant respecte RentalAgreement un cop passa a ser OpenRental, ja que si no, perdriem informació dels descomptes aplicats en una reserva si es canvia la relació entre CarModel i CarGroup. Per tant en l'esquema faria falta pot ser un redefines o bé no ser una relació derivada. <i>DE:CreateCarGroup</i>

Tipus de relacions

Nom	P1	Tipus	P2	Tipus	Motiu
LocatedAt	Branch	Constant	Country	Permanent	4. Una Branch sempre es situa a un mateix lloc, no es 'mou'. <i>DE:CreateBranch</i>
/isAvailable	Branch	Variable	OwnCar	Variable	<ul style="list-style-type: none"> Un car no sempre pertany a la mateixa branch.
IsResponsibleFor	Branch	Variable	Car	Variable	<ul style="list-style-type: none"> Un car no sempre pertany a la mateixa branch. <i>DE:TransferOwnership</i>
Orders	Branch	Variable	PendantCarOrder	constant	<ul style="list-style-type: none"> PendantCarOrder ho hem considerat Variable. PendantCarOrder les dades per dur-la a terme no han de ser modificables <i>DE:OrderCar</i>
OfModel	PendantCarOrder	Constant	CarModel	Variable	<ul style="list-style-type: none"> PendantCarOrder ho hem considerat Variable. PendantCarOrder les dades per dur-la a terme no han de ser modificables <i>DE:OrderCar</i>
IsBranchType	Branch	Constant	BranchType	Permanent	<ul style="list-style-type: none"> No es menciona res ni sembla coherent que canviï de tipus una branch. <i>DE:CreateBranch</i>

Reificacions

Nom	P1	Tipus	P2	Tipus	Tipus d'entitat	Motiu
/CountryBrachType	Country	Permanent	Branch	Constant	Permanent	Una brach mai es mou de lloc, i al ser un tipus d'entitat permanent, per la propietat 1, country és permanent.
ConcreteIndicator	/CountryBranchType	Variable	PerformanceIndicator	Variable	Variable	El sistema seguiria complint els requisits del sistema si la població d'aquest tipus d'entitat és variable. <i>DE: DefineTargetForConcreteIndicator.</i>
TransferAgreement	Branch	Variable	Branch	Variable	Variable	Hi han dos DE, un que crea i un altre que elimina entitats d'aquest tipus. <i>DE:CreateTransferAgreement, CancelTransferAgreement</i>

Nom	P1	Tipus	P2	Tipus	Tipus d'entitat	Motiu
/ModelAvailability	Branch	Variable	CarModel	Variable	Variable	Els cars dels que disposa una branch no són sempre els mateixos. Per tant, els models tampoc tenen perquè ser els mateixos.
/DemandXModel	Branch	Variable	CarModel	Variable	Variable	Són entitats resultants d'un càlcul de les reserves del dia següent. Cada dia desapareixen unes quantes.
/DemandXGroup	CarGroup	Variable	Branch	Variable	Variable	Són entitats resultants d'un càlcul de les reserves del dia següent. Cada dia desapareixen unes quantes.
CarGroupQuota	CarGroup	Variable	Branch	Variable	Variable	No es donen de baixa les instàncies de les relacions amb Branch, però no presenta cap problema poder-les eliminar. <i>DE:CreateCarGroup.</i>
/GroupAvailability	CarGroup	Variable	Branch	Variable	Variable	Són els grups dels models de cotxes disponibles actualment per la sucursal. Pot ser que tot un grup de models de cotxes no estigui disponible.

Atributs

Nom	Tipus d'entitat on apareix	Tipus	Motiu
Name	Branch	Variable	No sembla lògic afegir cap restricció. <i>DE:CreateBranch</i>
Name	BranchType	Constant	És absurd poder canviar-ho, pel tipus d'atribut que és
Name	Country	Variable	Pot donar-se el cas que per qüestions polítiques el país canviï de nom. <i>DE:RecordCountryInformation</i>
MechanicalConditionsRequirements	Country	Variable	Poden canviar ja que les lleis imposades en la country canvien <i>DE:RecordCountryInformation</i>
EmissionsRequirements	Country	Variable	Poden canviar ja que les lleis imposades en la country canvien <i>DE:RecordCountryInformation</i>
CarTax	Country	Variable	Poden canviar ja que les lleis imposades en la country canvien <i>DE:RecordCountryInformation</i>
CurrentMileage	OwnCar	Variable	Cada cop que un car arriba de ser llogat, el nombre de quilòmetres recorreguts enregistrats han d'actualitzar-se. <i>DE:ReceiveCar, RecordNewMileage</i>
MilageFromLastService	OwnCar	Variable	Cada cop que un cotxe passa pel manteniment, aquesta xifra s'actualitza. <i>DE:ReceiveCar, EndOfMaintenance</i>
LastMaintenanceDate	OwnCar	Variable	Cada cop que un cotxe passa pel manteniment, aquesta xifra s'actualitza. <i>DE:ReceiveCar, EndOfMaintenance</i>
acquisitionDate	OwnCar	Constant	Un car només es rep en una data. <i>DE:ReceiveCar</i>
/available?	OwnCar	Variable	Un car en certes condicions pot passar de disponible a no disponible i al revés.
/assigned	OwnCar	Variable	Si està ja assignat a alguna reserva
RegistrationNumber	Car	Constant	És un valor fixat sense semàntica que es fixat quan és rep un cotxe. <i>DE:ReceiveCar</i>
Id	PendantCarOrder	Constant	Una comanda és fa un cop i és la que és. <i>DE:OrderCar</i>
Name	CarModel	Constant	Un model de cotxe sempre conserva el mateix nom. <i>DE:CreateCarModel</i>
Carachteristics	CarModel	Variable	No sembla lògic afegir cap restricció, poden afegir-se, treure's, de tots els cotxes d'un model, exemple: aire acondicionat. <i>DE:CreateCarModel</i>
/demand	DemandXModel	Variable	Pot canviar si es fan més reserves.

Nom	Tipus d'entitat on apareix	Tipus	Motiu
/quantity	/ModelAvailability	Variable	Si es canvien els models de grup
/quantity	GroupAvailability	Variable	Si es canvien els models de grup
Quota	CarGroupQuota	Variable	No sembla lògic afegir cap restricció <i>DE:ChangeCarGroupQuota.</i>
/demand	DemandXGroup	Variable	
Distance	TransferAgreement	Variable	No sembla lògic afegir cap restricció <i>DE:IntroduceTransferData, ChangeTransferAgreementData</i>
ExpectedTime	TransferAgreement	Variable	No sembla lògic afegir cap restricció <i>DE:IntroduceTransferData, ChangeTransferAgreementData</i>
Name	PerformanceIndicator	Variable	No sembla lògic afegir cap restricció. <i>DE:CreateIndicator</i>
Description	PerformanceIndicator	Variable	No sembla lògic afegir cap restricció. <i>DE:CreateIndicator</i>
TargetValue	ConcreteIndicator	Variable	No sembla lògic afegir cap restricció <i>DE:ChangeTargetForConcreteIndicators</i>

RENTAL AGREEMENT

Tipus d'entitats

<i>Nom</i>	<i>Tipus</i>	<i>Motiu</i>
EU_RentPerson	Permanent	S'enregistren tots els clients, tant els que fan alguna reserva, com els que al final la cancel·len. Això és necessari per poder saber si es recomanable o no fer una reserva a un antic client. <i>DE:RecordCustomer</i>
Blacklisted	Permanent	Inclou tots aquells clients que no se'ls hi pot deixar reservar cap cotxe i aquesta condició ha de ser permanent. <i>DE:BlacklistCustomer</i>
Customer	Variable	Pel mateix motiu que RentalAgreement. <i>DE:MakeRental, CancelCurrentRental</i>
LoyaltyMember	Variable	En aquest grup pertanyen els 'privilegiats' que han realitzat cert nombre de reserves i que poden realitzar reserves amb punts. Un customer pot pertànyer i deixar de pertànyer a aquest tipus d'entitat, tan en el cas de deixar de complir les condicions necessàries com en el cas que hagi produït alguna incidència amb un cotxe reservat. <i>DE: JoinLoyaltyIncentiveScheme, CancelLoyaltyMembership, RecordDamages, CarReturn</i>
Driving License	Variable	Es poden actualitzar les dades de conducció d'un customer més d'un cop. <i>DE:RecordDriverData</i>

Tipus de relacions binàries

<i>Nom</i>	<i>P1</i>	<i>Tipus</i>	<i>P2</i>	<i>Tipus</i>	<i>Motiu</i>
Pick-up	RentalAgreement	Constant	Country	Variable	3. Es fixa quan es fa un RentalAgreement . 4. Un RentalAgreement no és permanent. <i>DE:MakeRental</i>
DropOff	RentalAgreement	Constant	Country	Variable	<ul style="list-style-type: none"> Es fixa quan es fa un RentalAgreement . Un RentalAgreement no és permanent. <i>DE:MakeRental</i>
/rentGroup	RentalAgreement	Variable	CarGroup	Variable	4. El grup que es permet no sempre és el mateix que el que s'assigna. 5. S'assigna un grup posterior a la reserva. Per tant el valor és variable. 6. Un RentalAgreement no és permanent. 7. Repetim que aquesta relació derivada hauria de ser constant quan un rentalAgreement passa a ser una openedRental.
Visits	RentalAgreement	Constant	Country	Variable	<ul style="list-style-type: none"> Es fixa només quan es fa un RentalAgreement Un RentalAgreement no és permanent. <i>DE:MakeRental</i>

Nom	P1	Tipus	P2	Tipus	Motiu
BestDurations	RentalAgreement	Permanent	CarGroupDurationPrice	Variable	<ul style="list-style-type: none"> CarGroupDurationPrice és permanent i per tant, pot aparèixer durant la vida del RentalAgreement un millor CarGroupDurationPrice.
InitEnding	RentalAgreement	Constant	DateTime	Variable	<ul style="list-style-type: none"> Es fixa només quan es fa un RentalAgreement Un RentalAgreement no és permanent. <i>DE:MakeRental</i>
/agreedEnding	RentalAgreement	Constant	DateTime	Variable	<ul style="list-style-type: none"> Queda fixat quan es crea el RentalAgreement Un RentalAgreement no és permanent. <i>DE:MakeRental</i>
Drive	EU_RentPerson	Variable	RentalAgreement	Permanent	<ul style="list-style-type: none"> No s'ha de restringir el canvi de conductors mentre el RentalAgreement no és OpenedRental. Un RentalAgreement no és permanent. <i>DE:AddDriverToRental</i>
/hasFaults	EU_RentPerson	Constant	FaultSeriousness	Permanent	<ul style="list-style-type: none"> Un FaultsSeriousness és un accident que un client ha tingut en una ClosedRental. Sempre afectarà al mateix client, al de la ClosedRental. Sempre que un client faci un ClosedRental podrà acumular-ne més.
/belongsTo	EU_RentPerson	Constant	Branch	Permanent	<ul style="list-style-type: none"> Un EU_RentPerson pertany a la primera branch on va fer la primera recollida d'un cotxe d'una reserva. Les reserves fetes són de població permanent. La relació entre la RentalAgreement i la pickUp branch és constant respecte RentalAgreement.
Has	EU_RentPerson	Variable	DrivingLicense	Constant	El DE indicat permet canviar la relació en qualsevol moment. <i>DE:RecordDriverData</i>

Tipus de relacions ternàries

/applicableDiscount	RentalAgreement	Variable	Discount	Variable	RentalDuration	Variable	Un RentalAgreement no sempre tracta amb el mateix CarGroup.
---------------------	-----------------	----------	----------	----------	----------------	----------	---

Reificacions

Nom	P1	Tipus	P2	Tipus	Tipus d'entitat	Motiu
RentalAgreement	DateTime	Variable	Customer	Variable	Variable	Si un client està desaccord amb les decisions preses pot cancel·lar la reserva (CancelCurrentRental), on desapareixen les entitats d'aquest tipus. <i>DE:MakeReservation, CancelCurrentRental, MakeWalkInRental, MakeReservationWithCarModel, MakeWalkInRentalWithCarModel</i>

Nom	P1	Tipus	P2	Tipus	Tipus d'entitat	Motiu
AssignedCar	RentalAgreement	Permanent	Car	Permanent	Permanent	Passa de no tenir-ne a tenir-ne, quan es tracta d'aconseguir un cotxe pel dia de la reserva. <i>DE:CarAllocationWithAnExtremisRule, CarAllocationAutomatic, CarAllocationExtremisOption, mirar MakeReservation</i>

Atributs

Nom	Tipus d'entitat on apareix	Tipus	Motiu
/basicPrice	RentalAgreement	Variable	El preu basic depèn de les duracions que hi hagi, que es poden afegir o modificar.
/bestPrice	RentalAgreement	Variable	El millor preu, el qual també depèn de RentalDurations que hi hagi
/lastModification	RentalAgreement	Constant	Dia de la reserva.
/onRentInterval	RentalAgreement	Constant	Distància temporal entre el dia que s'agafa el cotxe fins que es retorna.
Name	EU_RentPerson	Variable	No sembla lògic afegir cap restricció <i>DE:RecordCustomer</i>
Id	EU_RentPerson	Variable	No té cap semàntica, i per tant no hi ha cap mena de circumstància per la qual hauria de canviar. <i>DE:RecordCustomer</i>
BirthDate	EU_RentPerson	Constant	Una persona només té una data de naixement. <i>DE:RecordCustomer</i>
Address	EU_RentPerson	Variable	Una persona pot canviar de lloc de residència <i>DE:RecordCustomer</i>
Telephone	EU_RentPerson	Variable	És provable que una persona canviï de número de telèfon. <i>DE:RecordCustomer</i>
Number	DrivingLicense	Constant	El número de llicència és un identificador, no hi és dona el cas que es pugui canviar en el món real. <i>DE:RecordDriverData</i>
Issue	DrivingLicense	Constant	És una dada que donada una llicència no pot donar-se cap circumstància en la que canviï. <i>DE:RecordDriverData</i>
Expiration	DrivingLicense	Variable	Al renovar-se una llicència, hi ha una nova data de caducitat. <i>DE:RecordDriverData</i>
MembershipDate	LoyaltyMember	Constant	És la data en la que ha esdevingut un client en aquest grup. Per tant no pot canviar, ja que si nò, significaria que en el

Nom	Tipus d'entitat on apareix	Tipus	Motiu
			dia anterior no ho era. <i>DE:JoinLoyaltyIncentiveScheme</i>
/availablePoints	LoyaltyMember	Variable	Càlcul dels punts disponibles, els quals depenen de les reserves.
ExpectedPreparedTime	AssignedCar	Variable	És un valor que pot canviar degut a inconvenients que puguin sorgir
ActualTime	Prepared	Constant	Instant en el que ja ha estat llest. No canvia.

RENTAL AGREEMENT SUBCLASSES

Tipus d'entitats

<i>Nom</i>	<i>Tipus</i>	<i>Motiu</i>
/CarDamage	Constant	Correspon a un tipus de BadExperience concret, i BadExperience és constant.
BadExperience	Constant	S'assumeix que estan tots els valors de BadExpType

Tipus de relacions

<i>Relació</i>	<i>P1</i>	<i>Tipus</i>	<i>P2</i>	<i>Tipus</i>	<i>Motiu</i>
RequestedGroup	Reservation	Constant	CarGroup	Variable	<ul style="list-style-type: none"> Sorgeix només quan es dona d'alta una reserva. Reservation no és permanent.
RequestedModel	Reservation	Constant	CarModel	Variable	<ul style="list-style-type: none"> Sorgeix només quan es dona d'alta una reserva. Reservation no és permanent.
/NextDayReservations	Reservation	Variable	Branch	Variable	Les reserves poden ser eliminades.
ReturnedTo	ClosedRental	Constant	Branch	Permanent	Representa la relació entre un OpenRental i la branch en la que es retorna el cotxe, i per tant quan l'OpenRental esdevé tancat. Les dades relatives al tancament d'un OpenedRental no haurien de poder canviar, s'haurien de fixar en el mateix tancament.
ReturnedAt	ClosedRental	Constant	DateTime	Permanent	Representa la relació entre un OpenRental i la data en la que es retorna el cotxe, i per tant quan l'OpenRental esdevé tancat.

Reificacions

<i>Nom</i>	<i>P1</i>	<i>Tipus</i>	<i>P2</i>	<i>Tipus</i>	<i>Tipus d'entitat</i>	<i>Motiu</i>
ClosedRental	DateTime	Permanent	Customer	Permanent	Permanent	Un cop el lloguer finalitza, es diu que el RentalAgreement està tancat. Un cop tancat es manté en aquell estat sempre <i>DE:CarReturn</i>
/LateReturn	DateTime	Permanent	Customer	Permanent	Permanent	Degut a que només es la condició de mirar si el cotxe ha arribat més tard de la data prevista. Això és un fet que no es pot canviar. <i>DE:CarReturn</i>
/EarlyReturn	DateTime	Permanent	Customer	Permanent	Permanent	Degut a que només es la condició de mirar si el cotxe ha arribat més aviat de la data prevista. Això és un fet que no

Nom	P1	Tipus	P2	Tipus	Tipus d'entitat	Motiu
						es pot canviar. <i>DE:CarReturn</i>
/PaidWithPointsRental	DateTime	Permanent	Customer	Permanent	Permanent	És el cas que hagi estat pagada per un customer del LoyaltyMember mitjançant punts. Torna a ser un fet que no es pot canviar.
CanceledReservation	DateTime	Permanent	Customer	Permanent	Permanent	Una reserva quan passa a estar cancel·lada, no canvia d'estat. A més, són especialment interessants per saber els motius de perquè un customer està en la llista negra (CanceledCompany). <i>DE:CancelCustomersReservation, CancelReservation, FreeCarsInNotPickedUpR, CancelNoShowReservations</i>
CanceledCustomer	DateTime	Permanent	Customer	Permanent	Permanent	Una reserva cancel·lada per un usuari, mai canvia a ser cancel·lada per un altre factor. <i>DE:CancelReservation</i>
CanceledCompany	DateTime	Permanent	Customer	Permanent	Permanent	Una reserva cancel·lada per una empresa, mai canvia a ser cancel·lada per un altre factor. <i>DE:CancelCustomersReservation, CancelNoShowReservations, FreeCarsInNotPickedUpR</i>
ReservationWithSpecialDiscount	DateTime	Variable	Customer	Variable	Variable	Es pot repetir el DE ChooseDiscountOption, i fer canviar de tipus d'entitat una reserva.
PointsPaymentReservation	DateTime	Variable	Customer	Variable	Variable	El mateix motiu que l'anterior. <i>DE:ChooseDiscountOption</i>
Reservation	DateTime	Variable	Customer	Variable	Variable	Pel mateix motiu que RentalAgreement. Encara que això recull la idea d'un lloguer reservat prèviament. <i>DE:MakeReservation, CancelCurrentRental, MakeReservationWithCarModel,</i>
OpenedRental	DateTime	Permanent	Customer	Permanent	Permanent	Es diu que una renta és oberta quan han vingut en algun moment a recollir el cotxe que s'havia sol·licitat de lloguer. Un cop recollit, ja s'ha confirmat i per tant ja no es pot esborrar la entitat de RentalAgreement. <i>DE:OpenRental</i>
FaultSeriousness	ClosedRental	Permanent	BadExperience	Permanent	Permanent	Un OpenRental, quan passa a estar tancat, el cotxe pot haver estat tornat tard, sense pagar o bé amb danys. Els problemes de tancament d'una entitat d' OpenRental reben aquest nom. Es fa en diferent instant que en tancar un RentalAgreement (CarReturn). <i>DE: DefaultingCustomer</i>
ExtendedRental	DateTime	Permanent	Customer	Peermanent	Permanent	Un cop s'ha extés un lloguer més dels dies previstos, sempre serà un lloguer extés. <i>DE:RentalExtension</i>

Nom	P1	Tipus	P2	Tipus	Tipus d'entitat	Motiu
Extension	DateTime	Variable	ExtendedRental	Variable	Variable	Corresponent a les extensions dels dies que s'hagi fet del lloguer. En cas de que existeixi una relació lastNewEnding, es crea una extensió diferent, amb diferent data (DateTime) i extensionDone. <i>DE:RentalExtension</i>
DamageCost	/CarDamage	Permanent	ClosedRental	Permanent	Permanent	El cost dels danys del cotxe. Es fa en diferent instant que en tancar un RentalAgreement (CarReturn). <i>DE:RecordDamages</i>
/GuaranteedCanceled	DateTime	Permanent	Customer	Permanent	Permanent	Pel mateix motiu de que si un rentalAgreement passa a ser cancel·lat per sempre més ho és. A més GuaranteedReservation també és permanent.
GuaranteedReservation	DateTime	Permanent	Customer	Permanent	Permanent	Crec que no té sentit que una reserva deixi de ser garantitzada <i>DE:CancelGuaranteedReservation, GuaranteeReservation</i>

Atributs

Nom	Tipus d'entitat on apareix	Tipus	Motiu
PaymentType	ClosedRental	Constant	PaymentData (UC: Return of a car: Ho fa en dos passos i no pot ser)
CreditCardNumberDamages	ClosedRental	Constant	PaymentData (Igual que l'anterior)
/rentalPriceWithTax	ClosedRental	Constant	Els preus es gestionen per dates, per tant sempre és calcularà el millor per quan una entitat sigui del tipus ClosedRental.
Degree	FaultSeriousness	Variable	Un cop examinat el cotxe el grau de danys ha de ser sempre el que s'ha avaluat. Encara que sembla molt restrictiu no permetre'l canviar. <i>DE:DefaultingCustomer, recordDamages</i>
Price	DamageCost	Constant	recordDamagees
/extraInterval	/LateReturn	Constant	Quan es retorna el cotxe, la longitud de dies que s'ha trigat en tornar-ho sempre serà la mateixa.
/extraCostWithTax	/LateReturn	Constant	Quan es retorna el cotxe, el cost extra dels dies de retràs.
Type	BadExperience	Constant	Degut a que es l'atribut amb que s'identifica BadExperience i aquesta és constant

Nom	Tipus d'entitat on apareix	Tipus	Motiu
ActualPick-upTime	OpenedRental	Constant	El cotxe es recull una vegada <i>DE:OpenRental.</i>
ReservationDate	Reservation	Constant	La reserva té una data en la que es fa, i només aquesta. <i>DE:MakeReservation.</i>
CancellingDate	CanceledReservation	Constant	La reserva un cop és cancel·lada no deixa de ser-ho. No és intermitent. Per tant només té una data en la que és cancel·lada <i>DE:CancelCustomersReservation, CancelReservation, FreeCarsInNotPickedUpR.</i>
Motivation	<i>CanceledCompany</i> CanceledReservation	constant	El motiu de cancel·lació es fixa en el moment de cancel·lar la reserva. <i>DE:CancelCustomersReservation, FreeCarsInNotPickedUpR, CancelNoShowReservations, CancelReservation (pero aquesta última està malament perquè s'ha oblidat)</i>
/fine	/GuaranteedCanceled	Constant	La penalització en cas de que la reserva guaranteed hagi estat cancel·lada el mateix dia de recollida.
CreditCardNumber	GuaranteedReservation	constant	<i>DE:CancelGuaranteedReservation, FreeCarsInNotPickedUpR</i>
ExtensionDone	Extension	constant	<i>DE:RentalExtension</i>

CARS, DISCOUNTS AND ENUMERATIONS

Tipus d'entitats

Nom	Tipus	Motiu
RentalDuration	Permanent	El sistema actual no disposa d'un DE que permeti l'esborrat d'aquest tipus d'entitats, ni hi hagi un esborrat implícit per poder satisfer certes condicions de l'esquema. Es necessari per calcular el millor preu d'una reserva, recordem que sempre s'aplica el millor descompte des de que s'agafa el cotxe fins a que es deixa. <i>DE:RecordNewRentalDuration</i>
Discount	Permanent	Pel mateix motiu que en RentalDuration. <i>DE: RecordNewDiscount.</i>
ClosedDiscount	Permanent	Discount ho és i a més un descompte un cop és tancat mai deixa de ser-ho. <i>DE: CloseDiscount.</i>
ServiceDepot	Variable.	No hi ha cap DE associat. De totes maneres s'hauria de verificar que no s'esborrin entitats relacionades amb cap MaintenanceScheduled, o bé considerar-la de població permanent.
RepairsScheduled	Variable	Un cotxe entra i surt d'una reparació. <i>DE: RecordDamages, EndOfRepairs</i>
BeingTransferredCar	Variable	Un cotxe entra a estar transpassant-se fins que arriba a la branch que l'ha sol·licitat. <i>DE: RequestTransfer, DoTransfer.</i>
ToBeSoldCar	Permanent	Inclou com a població els cotxes que EU-Rent Car considera que ja són vells , estan molt usats , necessita vendre'ls o ja han estat venuts. <i>DE: SellCarsInNeed, SellCar</i>
SoldCar	Permanent	Té com a instàncies els cotxes que han estat venuts ja, i que han format part en algun moment com a cotxes de lloguer de l'empresa. Conforma l'estat final d'un cotxe en el sistema, i per tant les entitats d'aquest tipus mai desapareixen. <i>DE: ConfirmCarSale</i>
MaintenanceScheduled	Variable	Una entitat de Car és d'aquest tipus quan li toca el període de revisió. Aquest període té un inici i un final, per tant, les entitats de Car poden pertànyer i deixar de pertànyer a aquest tipus. <i>DE:ScheduleMaintenance, EndOfMaintenance</i>
/NeedMaintenance	Variable	Justificat per MaintenanceScheduled. <i>DE:EndOfMaintenance</i>
/NeedToBeSoldCar	Permanent	Són els cotxes vells o molt usats. Un cotxe es posa en venda quan passa a complir aquestes condicions, i per tant els cotxes venuts també les mantenen. Aquest tipus de cotxes no desapareixen del sistema.

DataTypes

Nom
Duration
PayType
BadExpType
BranchTypeName

Tèsi de Màster

Restriccions d'integritat temporals constant i permanent

Raquel Pau

Universitat Politècnica de Catalunya

Nom
Period
Level
CancellingMotivation
MaintenanceRequirements
DateTime
String
Double
Boolean
String
Integer
Natural
Object
Duration
Money
Time
Date

Tipus de relacions

Relació	P1	Tipus	P2	Tipus	Motiu
/In	MaintenanceSheduled	Constant	ServiceDepot	Variable	Encara que la Brach canviï de ServiceDepot, la reparació, hauria de fer-se en el que existia quan va aparèixer la reparació.
Serves	ServiceDepot	Variable	Branch	Variable	És natural deixar que es pugui canviar de ServiceDepot. No s'ha trobat cap DE associat a aquesta relació, ni el CreateBranch.
TransfereedTo	BeingTransferredCar	Constant	Branch	Variable	Quan s'està transferint un cotxe, no hi ha canvis d'itinerari. És perquè ha estat sol·licitat. <i>DE: RequestTransfer</i>
IsOfA	CarModel	Permanent	Car	Constant	<ul style="list-style-type: none"> Un cotxe no canvia de model. (constant) Cada cop que l'empresa rep un cotxe, aquest tindrà un model. (permanent)

Relació	P1	Tipus	P2	Tipus	Motiu
					<i>DE: ReceiveCar</i> (pero no ho fa)
/isGroup	Car	Variable	CarGroup	Variable	Depèn del grup del model, i aquest es pot canviar <i>DE: ChangeModelGroup</i>
Longer	RentalDuration	Variable	RentalDuration	Variable	Les cardinalitats no estan clares. Però si la semàntica és establir un ordre de duració de lloguers per preus. Al donar d'alta duracions, aquest ordre pot canviar. <i>DE: RecordNewRentalDuration</i>
ApplicableDurations	RentalDuration	Permanent	Discount	Permanent	Per poder calcular el millor descompte. <i>DE: RecordNewDiscount</i>

Reificacions

Nom	P1	Tipus	P2	Tipus	Tipus d'entitat	Motiu
CarGroupDurationPrice	CarGroup	Permanent	RentalDuration	Permanent	Permanent	Per mantenir un historial de les duracions possibles de preu pels diferents grups de cotxes i així conèixer els possibles que es van poder aplicar a les reserves. Si no podria ser que degut a les condicions de pagar, les quals tenen en compte les ofertes de preus abans de agafar el cotxe i després de deixar-ho, agafant la més econòmica no es podria realitzar, i per tant deixaria de conèixer-se el preu de la reserva.
EndDurationPrice	CarGroup	Permanent	RentalDuration	Permanent	Permanent	Pel motiu anterior.

Atributs

Atribut	Tipus d'entitat on apareix	Tipus	Motiu
Name	ServiceDepot	Variable	Pot canviar de nom el local. No DE
Capacity	ServiceDepot	Variable	Pot ampliar o reduir la seva capacitat. No DE
Price	CarGroupDurationPrice	Variable	<i>DE: ChangeCarGroupDurationPrice</i>
Name	Discount	Variable	Per qüestions de màrketig pot canviar de nom. <i>DE: RecordNewDiscount.</i>
Effect	Discount	Constant	Si canvia l'efecte, canvia el descompte en si. <i>DE: RecordNewDiscount.</i>
Description	Discount	Variable	Canvi d'intencionalitats, motius... <i>DE: RecordNewDiscount.</i>
BeginningDate	Discount	Constant.	Per evitar canviar la data d'inici un cop aplicat el descompte. Podria ser variable, però llavors faria falta una restricció que

Atribut	Tipus d'entitat on apareix	Tipus	Motiu
			no es pogués canviar un cop feta alguna reserva amb possibilitat d'aplicar dit descompte. <i>DE: RecordNewDiscount.</i>
ReservationTime?	Discount	Variable	No s'entén la semàntica
EndingDate	ClosedDiscount	Constant	El mateix cas que BeginningDate. Per evitar que no es pogués canviar un cop feta alguna reserva amb possibilitat d'aplicar dit descompte <i>DE:CloseDiscount</i>
BeginningDate	RepairsSheduled	Constant	Segons el DE RecordDamages scheduleReparations dona la data, però no està definida. Per sentit comú hauria de ser now().
DisposalDate	SoldCar	Constant	La data en la que es ven i aquesta no canvia. DE: ConfirmCarSale.
EndingDate	EndDurationPrice	Variable	Es pot estendre la duració del CarGroupDurationPrice.

7.2 Cas d'estudi Ebay

USUARIS, ARTICLES I CATEGORIES

Tipus d'entitats

<i>Nom</i>	<i>Tipus de població</i>	<i>Motiu</i>
Pre-Usuari	Permanent	Les seves instàncies són tots els usuaris, tant aquells que ja han fet una compra com aquells que només han enregistrat les seves dades i encara no les han confirmat. Degut a que en qualsevol moment es poden confirmar les dades, un cop un visitant les enregistra, aquelles dades sempre pertanyeran a un pre-Usuari.
Usuari	Permanent	Per poder obtenir informació històrica respecte a la oferta i la demanda, cal que les dades de l'usuari mai s'eliminin. Per tal de diferenciar els usuaris actius dels que no, hi ha un subtipus anomenat <i>UsuariNoActiu</i> . Per tant, això permet que una entitat usuari mai sigui eliminada de la base d'informació.
UsuariNoActiu	Permanent	Un cop es dona de baixa un usuari, no pot tornar a ser usuari. Si es torna a registrar, possiblement tingui les mateixes dades personals però el seu identificador canviï
Venedor	Permanent	Representa tots els usuaris que han estat venedors algun cop en el sistema. Llavors, un cop un usuari esdevé venedor, mai deixa de ser-ho encara que no sigui actiu. El motiu entre d'altres coses es per conèixer les dades d'un article ofert en algun moment, ja que aquests són permanents. No té de sentit tenir una operació que elimini un venedor.
Notificació	Permanent	Hi han operacions que treballen amb tipus concrets de Notificacions, com per exemple, el cas de poder rebre un avís que hi ha una segona oportunitat sobre l'oferta d'un article. Inicialment, hi han aquelles entitats que són imprescindibles pel bon funcionament del sistema. Es considera que es provable que en el futur es vulguin afegir noves notificacions, encara que actualment no hi hagi cap cas d'ús que les afegeixi.
Article	Variable	Un article es podria eliminar de la base d'informació si deixa de ser ofert per tots els venedors i que no hi ha cap oferta sobre aquest. Pot ser són massa comprovacions, però es poden inferir respecte altres restriccions temporals de tipus constant i permanent dels tipus d'entitats amb les que es relaciona i del tipus de participació d'aquestes en les relacions. Concretament : Venedor és permanent La participació de venedor en propietari és constant ArticleOfert és permanent La participació en "posar en venda" d'ArticleOfert és constant IMPORTANT: Acabem de veure un cas que podríem arribar a tenir redundància
Categoria	Variable	Un administrador pot eliminar categories, sempre i quan no continguin articles les categories finals del subarbre. Operació: Eliminar categoria.
CategoriaFinal	Variable	Pel mateix motiu que Categoria
CategoriaNoFinal	Variable	Pel mateix motiu que Categoria
/CategoriaPrincipal	Variable	Pel mateix motiu que Categoria
Cerca	Permanent	Qualsevol cerca s'enregistra per saber les cerques més populars de cada categoria. Degut a que no es coneix l'antiguitat de les cerques, no es pot saber si estem tractant amb una cerca recent o no. No hi ha cap operació que elimini les cerques. La única que ho podria fer és l'operació d'eliminar categoria, però no comprova si hi ha cerques associades per eliminar-les, així que només s'elimina la relació amb categoria i la entitat de cerca queda en el sistema.

DataTypes

<i>Nom</i>
DadesBancaries
TipusNotificació
TipusCerca
Divisa
String
Data
Integer
Boolean
Import

Tipus de relacions

<i>Nom</i>	<i>P1</i>	<i>Tipus</i>	<i>P2</i>	<i>Tipus</i>	<i>Motiu</i>
Preferències notificació	Usuari	Variable	Notificació	Variable	Es poden definir les preferències de Notificacions en qualsevol instant, i modificar-les, encara que en el treball esta mal indicat en el contracte de l'operació escollir-notificacions
DinsDe	Cerca	Variable	Categoria	Permanent	Hi han cerques que es fan sobre categories. Cerca és permanent tal i com ja s'ha comentat. Quan s'elimina una categoria no es comprova que hi hagin cerques associades, per tant aquestes poden quedar sense estar relacionades amb cap categoria(Variable). D'altra banda, una categoria cada cop té més cerques associades. (Permanent)
Relacionades	/CategoriaPrincipal	Variable	Categoria	Variable	Dos categories són relacionades amb l'operació relacionar-categories. Es pot donar en qualsevol moment modificant la relació anterior.
Categories preferides	Usuari	Variable	CategoriaFinal	Variable	Les preferències de l'usuari poden canviar
Propietari	Venedor	Variable	Article	Constant	Un venedor pot ser li interessa desprendre's d'un article del qual no oferirà ofertes. Un article no es pot transferir d'un venedor a un altre.
Pertany a	Article	Variable	CategoriaFinal	Variable	La participació de CategoriaFinal és variable perquè a un venedor pot ser li interessa desprendre's d'un article del qual no oferirà ofertes. D'altra banda, la participació d'Article és variable, perquè aquests es poden moure de categoria.
Es descomposa en	Categoria	Variable	CategoriaNoFinal	Variable	Les categories encara que no existeixi cap operació per moure's les categories com si fossin carpetes d'un SO, sembla bastant natural permetre-ho.

Cerques Preferides	Usuari	Variable	Cerca	Variable	Un usuari pot canviar de preferències de cerques, encara que no es presenti l'operació que ho permeti.
/Té	Cerca	Variable	Categoria	Variable	Les cerques més freqüents canvien al llarg del temps.

Reificacions

Nom	P1	Tipus	P2	Tipus	Tipus d'entitat	Motiu
NotificacióCerca	Usuari	Variable	Cerca	Variable	Variable	Si una notificació de cerca caduca, s'elimina del sistema al acabar el dia.

Atributs

Nom	Tipus d'entitat on apareix	Tipus	Motiu
idUsuari	Pre-Usuari	Variable	Amb l'operació ' <i>canviarIdentificador</i> ' es pot modificar el valor d'aquest atribut.
Password	Pre-Usuari	Variable	Amb l'operació ' <i>nou-pwd</i> ' es pot modificar el valor d'aquest atribut
Email	Pre-Usuari	Variable	Amb l'operació ' <i>nou-mail</i> ' es pot modificar el valor d'aquest atribut.
Nom	Pre-Usuari	Variable	Amb l'operació ' <i>noves-dades</i> ' es pot modificar el valor d'aquest atribut.
Cognom	Pre-Usuari	Variable	Amb l'operació ' <i>noves-dades</i> ' es pot modificar el valor d'aquest atribut.
preguntaSecreta	Pre-Usuari	Variable	Ha de ser modificable si es canvia el password
respostaSecreta	Pre-Usuari	Variable	Ha de ser modificable si es canvia el password
dataNaixement	Pre-Usuari	Constant	Un usuari té una data de naixement i és única.
Telefon	Pre-Usuari	Variable	Amb l'operació ' <i>noves-dades</i> ' es pot modificar el valor d'aquest atribut
Adreça	Pre-Usuari	Variable	Amb l'operació ' <i>noves-dades</i> ' es pot modificar el valor d'aquest atribut
Ciutat	Pre-Usuari	Variable	Amb l'operació ' <i>noves-dades</i> ' es pot modificar el valor d'aquest atribut
CodiPostal	Pre-Usuari	Variable	Amb l'operació ' <i>noves-dades</i> ' es pot modificar el valor d'aquest atribut
Regió	Pre-Usuari	Variable	Amb l'operació ' <i>noves-dades</i> ' es pot modificar el valor d'aquest atribut
Pais	Pre-Usuari	Variable	Amb l'operació ' <i>noves-dades</i> ' es pot modificar el valor d'aquest atribut
codiConfirmació	Pre-Usuari	Variable	Amb l'operació ' <i>nou-mail</i> ' es pot modificar el valor d'aquest atribut
tarjaCredit	Usuari	Variable	Un usuari pot desitjar canviar de tarja de crèdit, encara que no hi hagi cap operació actualment que ho permeti.
idVerificat	Usuari	Variable	Amb l'operació ' <i>verificar-id</i> ' es pot modificar el valor d'aquest atribut.

dataVerificació	Usuari	Variable	Amb l'operació ' <i>nou-id</i> ' es pot modificar el valor d'aquest atribut.
idAnterior	Usuari	Variable	Amb l'operació ' <i>nou-id</i> ' es pot modificar el valor d'aquest atribut.
dataCanvild	Usuari	Variable	Amb l'operació ' <i>nou-id</i> ' es pot modificar el valor d'aquest atribut.
canviMail	Usuari	Variable	Amb l'operació ' <i>nou-mail</i> ' es pot modificar el valor d'aquest atribut.
dataBaixa	UsuariNoActiu	Constant	Amb l'operació ' <i>baixa-usuari</i> ' es pot modificar el valor d'aquest atribut.
compteCorrent	Venedor	Variable	És natural canviar per motius personals de compteCorrent per recaudar les ventes, encara que no hi hagi cap operació actualment que ho permeti.
cobrarTaxes	Venedor	Variable	És natural canviar per motius personals el compteCorrent d'on cobrar els serveis, encara que no hi hagi cap operació actualment que ho permeti.
idArticle	Article	Constant	Representa un identificador del sistema per a qualsevol tipus d'article. Cap usuari ha d'interaccionar amb aquest valor, i no és natural que es canviï.
nomNotif	Notificació	Constant	Representa una de les possibles notificacions existents, no és un valor canviable, ja que sinó estariem representant una nova Notificació
dataSubscripció	NotificacióCerca	Constant	Representa la data en el que et subscrius a una cerca, en cas de que caduqui, has de subscriure't un altre cop.
/dataFi	NotificacióCerca	Constant	Sempre tindrà el mateix valor ja que és calculada afegint 90 dies a la data de subscripció, i aquesta última com ja hem vist és un atribut constant.
Paraules	Cerca	Constant	Degut a la naturalesa que presenta el concepte de cerca. És un element instantani, que es fa un cop, i que únicament serveix per accedir-hi com un element de consulta. Els seus camps no té sentit modificar-los perquè representaria una altra cerca.
Condicció	Cerca	Constant	Degut a la naturalesa que presenta el concepte de cerca, tal i com ja s'ha explicat
senseParaules	Cerca	Constant	Degut a la naturalesa que presenta el concepte de cerca, tal i com ja s'ha explicat
cercarATítol	Cerca	Constant	Degut a la naturalesa que presenta el concepte de cerca, tal i com ja s'ha explicat
NomesOfTancades	Cerca	Constant	Degut a la naturalesa que presenta el concepte de cerca, tal i com ja s'ha explicat
NomesVendalmm	Cerca	Constant	Degut a la naturalesa que presenta el concepte de cerca, tal i com ja s'ha explicat
NomesRegals	Cerca	Constant	Degut a la naturalesa que presenta el concepte de cerca, tal i com ja s'ha explicat
DiversesUnitats	Cerca	Constant	Degut a la naturalesa que presenta el concepte de cerca, tal i com ja s'ha explicat
aPartirDe	Cerca	Constant	Degut a la naturalesa que presenta el concepte de cerca, tal i com ja s'ha explicat
finsA	Cerca	Constant	Degut a la naturalesa que presenta el concepte de cerca, tal i com ja s'ha explicat
Regio	Cerca	Constant	Degut a la naturalesa que presenta el concepte de cerca, tal i com ja s'ha explicat
Pais	Cerca	Constant	Degut a la naturalesa que presenta el concepte de cerca, tal i com ja s'ha explicat
Moneda	Cerca	Constant	Degut a la naturalesa que presenta el concepte de cerca, tal i com ja s'ha explicat
/nombreCerques	Cerca	Variable	Si apareix una nova cerca, aquest valor canviarà pel conjunt de cerques que busquin la mateixa paraula

nomCat	Categoria	Variable	El valor d'aquest atribut es pot modificar ja amb l'operació ' <i>modificar-categoria</i> '
/numArticles	Categoria	Variable	Cada cop que s'afegeix un nou article, s'incrementa aquest valor
Caract	CategoriaFinal	Constant	Si hi ha articles relacionats, amb la categoria, passaria un desajust modificant el valor d'aquest atribut per causa del " <i>materializes</i> ".

OFERTA D'ARTICLES

Tipus d'entitats

<i>Nom</i>	<i>Tipus de població</i>	<i>Motiu</i>
ArticleOfert	Permanent	No té cap mena de significat esborrar les instàncies d'aquest tipus d'entitat si existeixen els subtipus de OfertaCancel·lada o OfertaTancada, que són els únics motius en els que podria eliminar-se una entitat d' <i>ArticleOfert</i> .
ArticleEnVenda	Permanent	Un article ofert o es referent a terrenys(ArticleAnunciat) o no(ArticleEnVenda). Un article no es pot canviar un cop es llança la oferta (ArticleOfert), no és natural.
ArticleAnunciat	Permanent	El mateix motiu que l'explicat per <i>ArticleEnVenda</i>
ArtVendaSubhasta	Permanent	"Si no hi ha hagut cap demanda i queden més de 12 hores, el venedor pot modificar qualsevol característica excepte el format de venda" Aquesta frase ens garanteix que no podem fer canvis de subtipus de ArticleOfert, pel que fa al tipus de venda. Per modificar el format de venda, s'ha de cancel·lar l'article ofert actual i crear un nou articleOfert.
ArtVendalmmediata	Permanent	El mateix motiu que l'explicat en <i>ArtAnunciatSubhasta</i> .
ArtAnunciatSubhasta	Permanent	El mateix motiu que l'explicat en <i>ArtVendaSubhasta</i> .
artVendaSubMultiple	Permanent	El mateix motiu que l'explicat en <i>ArtVendaSubhasta</i> .
ArtVendaSubEstandard	Variable	El mateix motiu que l'explicat en <i>ArtVendaSubhasta</i> .
Foto	Variable	Si una foto no utilitza cap ArticleOfert s'ha d'eliminar. Recordem que la CardMin(Foto;articleOfert;Mostra)=1
Extra	Constant	Les seves entitats són els diferents valors de TipusExtra
Moneda	Permanent	Si no fos permanent, eliminariem ofertes donat que la participació de <i>ArticleOfert</i> en el tipus de relació "En(p1:ArticleOfert, p2:Moneda)" és constant. El motiu pel qual existeix aquesta entitat també és per com canvia el valor de tipusCanvi en el temps.

DataTypes

<i>Nom</i>
TipusExtra
Periode
Real

Tipus de relacions

<i>Nom</i>	<i>P1</i>	<i>Tipus</i>	<i>P2</i>	<i>Tipus</i>	<i>Motiu</i>
Mostra	ArticleOfert	Variable	Foto	Variable	Es natural que una venedor canviï les fotos d'una oferta per atraure clients.
Té	ArticleOfert	Variable	Extra	Variable	Es natural que el venedor canviï els extres per a que el seu anunci tingui més

Nom	P1	Tipus	P2	Tipus	Motiu
					interessats.
En	ArticleOfert	Variable	Moneda	Variable	Amb l'operació ' <i>modificar-oferta</i> ' es poden modificar les instàncies d'aquesta relació
Observar	Usuari	Variable	ArticleOfert	Variable	Encara que no hi hagi cap operació que elimini instàncies d'aquest tipus de relació, és bastant natural pensar que un usuari li deixi d'interessar l'evolució d'una oferta.
Oferir	Article	Permanent	ArticleOfert	Constant	Un article ofert sempre ha de pertànyer al mateix article. També degut a que ArticleOfert té una població permanent, es compleix la permanència de l'altra participació.
S'ofereix A	ArticleOfert	Variable	CategoriaFinal	Variable	Amb l'operació ' <i>modificar-oferta</i> ' es poden modificar les instàncies d'aquesta relació.

Atributs

Nom	Tipus d'entitat on apareix	Tipus	Motiu
tempsInici	ArticleOfert	Variable	Amb l'operació ' <i>modificar-oferta</i> ' es pot modificar el valor d'aquest atribut.
numDies	ArticleOfert	Variable	Amb l'operació ' <i>modificar-oferta</i> ' es pot modificar el valor d'aquest atribut.
/tempsFi	ArticleOfert	Variable	Amb l'operació ' <i>modificar-oferta</i> ' es pot modificar el valor d'aquest atribut.
títol	ArticleOfert	Variable	Amb l'operació ' <i>modificar-oferta</i> ' es pot modificar el valor d'aquest atribut.
descripció	ArticleOfert	Variable	Amb l'operació ' <i>modificar-oferta</i> ' es pot modificar el valor d'aquest atribut.
tempsAlta	ArticleOfert	Constant	Representa l'instant en el que es va crear l'oferta, i aquest mai canvia.
localització	Foto	Variable	Una foto pot canviar de lloc
nomExtra	Extra	Constant	Representa a un ' <i>TipusExtra</i> ', en cas de canviar-ho representaria un altre, i per tant tindriem un fet incorrecte.
nom	Moneda	Constant	No pot canviar de divisa representant, no té sentit.
tipusCanvi	Moneda	Constant	Representa el tipus de Canvi d'una divisa en un instant de temps. No pot canviar.
dataActualització	Moneda	Constant	Representa la data en la qual el tipus de canvi d'una divisa ha canviat. No pot canviar.
preu	ArticleAnunciat	Variable	Encara que no hi hagi cap operació que ho permeti, sinó ha començat l'oferta és natural poder-ho canviar.
preuInicial	ArtVendaSubhasta	Variable	Hauria de ser variable sempre i quan no hagués començat l'oferta.
preuMinim	ArtVendaSubEstandard	Variable	Amb les operacions " <i>afegir-pv</i> " i " <i>eliminar-pm</i> " es pot modificar el valor d'aquest atribut.
numOferts	ArtVendaSubMultiple	Variable	Amb l'operació " <i>modificar-pm</i> " es pot modificar el valor d'aquest atribut.
preuVenda	ArtVendaImmediata	Variable	Amb l'operació " <i>modificar-pv</i> " es pot modificar el valor d'aquest atribut.
numOferts	ArtVendaImmediata	Variable	És natural poder canviar el nombre d'articles.

O F E R T A I D E M A N D A

Tipus d'entitats

<i>Nom</i>	<i>Tipus</i>	<i>Motiu</i>
OfertaTancada	Permanent	Representa una oferta que ja ha vençut. Aquesta pot haver tingut èxit o bé haver estat rebutjada perquè el venedor no accepta el preu proposat. Degut a que és una generalització de dos estats finals, la població d'aquest tipus d'entitat és permanent.
OfertaAcceptada	Permanent	És un estat final d'ArticleOfert, el qual recordem que és permanent.(Veure en diagrama de transició d'estats)
OfertaRebutjada	Permanent	El mateix motiu que OfertaAcceptada
/OfertaOberta	Variable	Cada dia potser caduquen algunes ofertes, passant a estar tancades. Per tant la seva població és variable.
/OfertaPendent	Variable	Cada dia hi han ofertes que es compleix que el dia que es comença a oferir l'oferta és el dia actual i per tant, deixen de pertànyer a aquest subtipus.
OfertaCancel·lada	Permanent	És un estat final d'una oferta, el qual es caracteritza per ser el venedor qui desitja cancel·lar-la.
Dita	Permanent	Degut a : la CardMin(Dita;DemCompraSubEstandard;AplicarTactica)=1 la participació en la relació AplicarTactica de SemCompraSubEstandard és permanent.
IncrementDites	Constant	No es pot canviar la taula d'increments de dites. Fixada a la pagina 142

DataTypes

Temps

Tipus de relacions

<i>Nom</i>	<i>P1</i>	<i>Tipus</i>	<i>P2</i>	<i>Tipus</i>	<i>Motiu</i>
/Guanyen	Demanda	Variable	ArticleOfert	Variable	Indica quin és el possible guanyador final en cada instant. Aquest varia en funció del temps.
Permetre	ArticleOfert	Variable	Usuari	Variable	Es pot modificar a qui permetre accedir a l'oferta en tot moment amb l'operació de "permetre".
Bloquejar	Venedor	Variable	Usuari	Variable	Es pot modificar a qui bloquejar per comprar articles oferts en tot moment amb l'operació de "bloquejar".
AplicarTàctica	DemCompraSubEstandard	Constant	Dita	Permanent	Només s'indica la dita quan es fa una demanda de subhasta estàndard.

Reificacions

Nom	P1	Tipus	P2	Tipus	P3	Tipus	Tipus d'entitat	Motiu
Demanda	Usuari	Permanent	Temps	Permanent	ArticleOfert	Permanent	Permanent	Es necessita constància de les demandes que ha fet un usuari de les quals ha estat guanyador. Per tant necessitem conservar la informació. També hi han cerques per conèixer el grau de confiança de l'usuari tal i com s'explica en la secció de "comentaris"
DemCancel·lada	Usuari	Permanent	Temps	Permanent	ArticleOfert	Permanent	Permanent	"L'usuari indica quina demanda vol retirar i el motiu pel qual ho vol fer. Aquesta no s'eliminarà físicament, sinó que quedarà marcada com a cancel·lada per a després poder consultar-ne els motius". Es marca com a cancel·lada una demanda amb l'operació "Cancel·dem-usr"
DemandaContacte	Usuari	Permanent	Temps	Permanent	ArticleOfert	Permanent	Permanent	Els subtipus de demandes es classifiquen segons el tipus de venda de l'article ofert que sol·liciten. Això comporta que només pertanyin a una, i pertanyin sempre. Per tant, els subtipus que parteixen del discriminant tipusDem, tots són permanents.
DemandaCompra	Usuari	Permanent	Temps	Permanent	ArticleOfert	Permanent	Permanent	Pel mateix motiu que DemandaContacte
DemContacteSubhasta	Usuari	Permanent	Temps	Permanent	ArticleOfert	Permanent	Permanent	Pel mateix motiu que DemandaContacte
DemCompraSubhasta	Usuari	Permanent	Temps	Permanent	ArticleOfert	Permanent	Permanent	Pel mateix motiu que DemandaContacte
DemCompralmediata	Usuari	Permanent	Temps	Permanent	ArticleOfert	Permanent	Permanent	Pel mateix motiu que DemandaContacte
DemCompraSubEstandart	Usuari	Permanent	Temps	Permanent	ArticleOfert	Permanent	Permanent	Pel mateix motiu que DemandaContacte
DemCompraSubMúltiple	Usuari	Permanent	Temps	Permanent	ArticleOfert	Permanent	Permanent	Pel mateix motiu que DemandaContacte

Nom	P1	Tipus	P2	Tipus	Tipus d'entitat	Motiu
SegonaOportunitat	Usuari	Permanent	OfertaAcceptada	Permanent	Permanent	Representa la informació relativa a la segona oportunitat que té el venedor en vendre l'article en cas de que el guanyador no finalment no compri l'article. No té sentit esborrar entitats d'aquest tipus.

Atributs

<i>Nom</i>	<i>Tipus d'entitat on apareix</i>	<i>Tipus</i>	<i>Motiu</i>
PreuMàxim	Demanda	Constant	Es defineix una vegada, per canviar-ho s'ha de fer una altra demanda.
NumItems	Demanda	Constant	Es considera que no es pot modificar cap valor d'un atribut d'una demanda, si un cas correspondria a fer una altra demanda.
NumItems	DemCompralmediata	Constant	Es considera que no es pot cap valor d'un atribut d'una demanda, si un cas correspondria a fer una altra demanda.
Motiu	DemCancel.lada	Constant	Un cop una demanda és cancel·lada no té sentit canviar-li valors.
Moment	DemCancel.lada	Constant	Un cop una demanda és cancel·lada no té sentit canviar-li valors.
Moment	Temps	Constant	Cada instant de temps es considera diferent entitat
/preuTotal	DeCompralmediata	Constant	Es calcula només a partir del preuVenda inicial de la demanda i del numItems. El preuVenda mai canviarà si ja té alguna demanda i el numItems ja hem vist que és constant.
/preuTotal	DemCompraSubMultiple	Variable	Es compten els items que et queden en prioritat a l'antiguitat de la demanda. Sempre seran els mateixos sempre que no es doni el cas de cancel·lar una demanda.
/numGuanyats	DemCompraSubMultiple	Variable	Depèn de que algun usuari no cancel·li la demanda per a que no et toquin pot ser més items.
Motiu	OfertaCancel·lada	Constant	No té sentit modificar el motiu de cancel·lació.
Moment	OfertaCancel·lada	Constant	És l'instant de temps pel qual s'ha cancel·lat l'oferta, per tant no pot canviar.
/preuActual	ArticleOfert	Variable	Aquest varia en funció de les dites de les subhastes o del preu màxim a oferir, etc..
PreuRebaixat	ArticleEnVenda	Variable	Amb l'operació " <i>modificar-pv</i> " es pot modificar el valor d'aquest atribut.
/preuFinal	OfertaAcceptada	Variable	Perquè es pot sol·licitar una segona oportunitat i això provoca un canvi en el seu valor.
/numResten	ArtVendalmediata	Variable	A mesura que hi ha més demandes, aquest valor decreix
/numResten	ArtVendaSubMultiple	Variable	Varia en funció de les unitats d'un mateix article ofert de les demandes guanyadores.
DesDe	Increment Dites	Constant	Estan donades per l'especificació
FinsA	Increment Dites	Constant	Estan donades per l'especificació
Increment	Increment Dites	Constant	Estan donades per l'especificació.

DESPRÉS DE LA VENDA: C O M E N T A R I S

Tipus d'entitats

No hi ha.

DataTypes

Puntuació

Tipus de relacions

<i>Nom</i>	<i>P1</i>	<i>Tipus</i>	<i>P2</i>	<i>Tipus</i>	<i>Motiu</i>
Fet per	Comentari	Constant	Usuari	Permanent	L'usuari que ha emès el comentari sempre és el mateix. Degut a que comentari és permanent tenim que la participació d'usuari és permanent.
Sobre	Comentari	Constant	Usuari	Permanent	L'usuari a qui afecta el comentari també sempre és el mateix. Degut a que el comentari és permanent, tenim que la participació d'usuari és permanent.

Refificacions

<i>Nom</i>	<i>P1</i>	<i>Tipus</i>	<i>P2</i>	<i>Tipus</i>	<i>Tipus d'entitat</i>	<i>Motiu</i>
Comentari	Temps	Permanent	OfertaAcceptada	Permanent	Permanent	Es desitja conèixer els tots els comentaris que ha fet o ha rebut un usuari.

Atributs

<i>Nom</i>	<i>Tipus d'entitat on apareix</i>	<i>Tipus</i>	<i>Motiu</i>
Valoració	Comentari	Constant	Un comentari és un element que es crea una vegada i ja no és modificable.
Text	Comentari	Constant	Un comentari és un element que es crea una vegada i ja no és modificable.
/feedback	Usuari	Variable	Depèn dels comentaris actuals en els que l'usuari sigui l'afectat.

TAXES

Tipus d'entitats

<i>Nom</i>	<i>Tipus</i>	<i>Motiu</i>
eBayTaxes	Permanent	Les taxes s'indiquen per cada any, o almenys és el que s'ha considerat com un període de temps raonable per a que no variïn. Per tant, cada any s'identifiquen taxes diferents.
TaxaInserció	Permanent	El mateix raonament que eBayTaxes.
TaxaPreuMinim	Permanent	El mateix raonament que eBayTaxes.
TaxaExtra	Permanent	El mateix raonament que eBayTaxes.
TaxaVenda	Permanent	El mateix raonament que eBayTaxes.
TaxaVendaAltres	Permanent	El mateix raonament que eBayTaxes.
TaxaVendaVehicles	Permanent	El mateix raonament que eBayTaxes.
TaxaInsercioGeneral	Permanent	El mateix raonament que eBayTaxes.
TaxaInsercioImmables	Permanent	El mateix raonament que eBayTaxes.
TaxaInsercioSubhasta	Permanent	El mateix raonament que eBayTaxes.
TaxaInsercioAnunci	Permanent	El mateix raonament que eBayTaxes.
TaxaInsercioVehicle	Permanent	El mateix raonament que eBayTaxes.

DataTypes

<i>Nom</i>
TipusVehicles
TipusTaxaInsl
PeriodeTaxaInslS
PeriodeTaxaInslA
TipusTaxaExtra
Any

Tipus de relacions

<i>Nom</i>	<i>P1</i>	<i>Tipus</i>	<i>P2</i>	<i>Tipus</i>	<i>Motiu</i>
/Se li aplica	ArticleOfert	Variable	EbayTaxes	Variable	La relació d'ArticleOfert amb Extra es variable, i afecta al càlcul de les relacions d'aquest tipus

Atributs

<i>Nom</i>	<i>Tipus d'entitat on apareix</i>	<i>Tipus</i>	<i>Motiu</i>
AnyTaxa	EbayTaxes	Constant	Els valor de qualsevol atribut d'una s'apliquen durant un any no es canvien com a molt aviat l'any següent. L'any que es canviï es crearà una de entitat nova del tipus corresponent amb les mateixes dades.
Taxa	TaxaInsercio	Constant	El mateix motiu que AnyTaxa.
Tipus	TaxaInsercioVehicles	Constant	El mateix motiu que AnyTaxa.
Tipus	TaxaInsercioImmables	Constant	El mateix motiu que AnyTaxa.
Dies	TaxaInsercioSubhasta	Constant	El mateix motiu que AnyTaxa.
Dies	TaxaInsercioAnunci	Constant	El mateix motiu que AnyTaxa.
De	TaxaInsercioGeneral	Constant	El mateix motiu que AnyTaxa.
A	TaxaInsercioGeneral	Constant	El mateix motiu que AnyTaxa.
De	TaxaPreuMinim	Constant	El mateix motiu que AnyTaxa.
A	TaxaPreuMinim	Constant	El mateix motiu que AnyTaxa.
Taxa	TaxaPreuMinim	Constant	El mateix motiu que AnyTaxa.
Categoria	TaxaExtra	Constant	El mateix motiu que AnyTaxa.
NomExtra	TaxaExtra	Constant	El mateix motiu que AnyTaxa.
Taxa	TaxaExtra	Constant	El mateix motiu que AnyTaxa.
De	TaxaVendaAltres	Constant	El mateix motiu que AnyTaxa.
A	TaxaVendaAltres	Constant	El mateix motiu que AnyTaxa.
Percentatge	TaxaVendaAltres	Constant	El mateix motiu que AnyTaxa.
Tipus	TaxaVendaVehicles	Constant	El mateix motiu que AnyTaxa.
Taxa	TaxaVendaVehicles	Constant	El mateix motiu que AnyTaxa.

Nom	Tipus d'entitat on apareix	Tipus	Motiu
/baseInsercio	ArticleOfert	Variable	Hi han aplicats en la regla de derivació atributs de naturalesa variable com preuVenda o preuMinim
/costInsercio	ArticleOfert	Variable	Depen del nombre de categories finals amb els que està relacionada, la qual cosa pot canviar amb modificar-oferta
/costPreuMinim	ArticleOfert	Variable	Es pot insertar una taxa de preu minim en qualsevol moment
/costVenda	ArticleOfert	Variable	En cas de que es tracti d'una TaxaVendaAltres s'emplea en la regla de derivació el nombre de d'items venuts fins al moment.
/costExtres	ArticleOfert	Variable	El nombre de fotos que es pot posar en una oferta pot variar. Per tant aquest cost també
/costTotal	ArticleOfert	Variable	Hi han subcostos tractats variables(tots).

7.3 Cas d'estudi DBLP

Tipus d'entitats

<i>Nom</i>	<i>Tipus</i>	<i>Motiu</i>
Person	Permanent	La seva població són els autors i editors de les publicacions enregistrades. Per cada publicació, si l'autor o editor no existeix, s'ha d'introduir a la base d'informació.
Publication	Permanent	La seva població són tots els llibres i revistes ja publicades (un cop són publicades mai deixen de ser-ho)
Book	Permanent	Les diferents nivells d'especialització de publicació tenen una classificació totalment estàtica. Per tant si Publication es permanent aquest subtipus també.
AuthoredPublication	Permanent	Idem a Book
EditedBook	Permanent	Idem a Book
AuthoredBook	Permanent	Idem a Book
BookChapter	Permanent	Idem a Book
JournalPaper	Permanent	Idem a Book
BookSection	Permanent	Els capítols d'un llibre es poden agrupar en seccions. La seva població engloba totes les seccions dels EditedBooks i BookSeriesIssue. Per tant, si aquests són permanents, cada cop que s'afegeix una instància d'aquests dos tipus d'entitats, poden haver-hi més booksections en la base d'informació
BookSeriesIssue	Permanent	La seva població són les diferents edicions d'un llibre. Com l'especialització de Book és totalment estàtica, al ser Book permanent, BookSeriesIssue també ho és
BookSeries	Permanent	Representa la col·lecció d'un llibre, si BookSeriesIssue és permanent, i aquest té participació constant, aquest és permanent. A part, té sentit que quan es doni d'alta un BookSeriesIssue si no existeix el BookSeries, es crei.
JournalSection	Permanent	De la mateixa manera que els llibres les revistes (Jorunal) es poden agrupar els papers en seccions. Llavors la població representa totes les seccions a les que pertanyen els diferents papers (els quals són publicacions). Per tant, cada cop que s'afegeixi una instància de JournalPaper, pot ser s'haurà d'afegir una JournalSection, que sempre serà la mateixa.
JournalIssue	Permanent	La seva població són les diferents edicions d'un volum d'una revista (Journal) en les que hi ha publicat algun JournalPaper. JournalPaper és un tipus d'entitat permanent, i per tant, JournalIssue també ho és.
ConferenceEdition	Permanent	La seva població són edicions de conferències (ex. JISBD-07) de les que hi ha alguna publicació. Si Publication és permanent, ConferenceEdition també.
JournalVolume	Permanent	La seva població són tots els volums d'una revista (Journal) en les que hi ha algun JournalPaper publicat. JournalPaper és un tipus d'entitat permanent, i per tant, JournalIssue també ho és.
Journal	Permanent	Són totes les revistes. En cas que una instància d'aquest tipus no hagi publicat res, es podrà eliminar
ConferenceSeries	Permanent	Són totes les conferències. En cas que una instància d'aquest tipus no hagi publicat res, es podrà eliminar

Tipus de relacions

<i>Nom</i>	<i>P1</i>	<i>Tipus</i>	<i>P2</i>	<i>Tipus</i>	<i>Motiu</i>
IsEditorOf	Person	Permanent	EditedBook	Constant	Una instància de Person s'afegeix, quan publica quelcom. Cada cop que publiqui un altre EditedBook, aquest sempre tindrà el mateix autor, i l'autor més EditedBooks.
--	EditedBook	Constant	BookSection	Constant	Un EditedBook sempre tindrà les mateixes seccions i una secció d'un llibre sempre pertanyerà al mateix llibre.
--	EditedBook	Constant	BookChapter	Constant	Un EditedBook sempre tindrà els mateixos capítols i un capítol d'un llibre sempre és referirà al mateix llibre.
--	EditedBook	Constant	ConferenceEdition	Constant	Un EditedBook podrà ser una publicació d'una ConferenceEdition i sempre serà la mateixa (encara que sigui cap). Una ConferenceEdition estarà en la BI si s'afegeix un EditedBook o un BookSeriesIssue o un JournalIssue com a publicació d'aquesta.
--	ConferenceSeries	Permanent	ConferenceEdition	Constant	Una ConferenceSeries, al llarg del temps fa ConferenceEditions, les quals s'enregistren en el sistema si publiquen quelcom. D'altra banda una ConferenceEdition sempre serà de la mateixa ConferenceSerie
--	ConferenceEdition	Constant	BookSeriesIssue	Constant	Anàleg a EditedBook i ConferenceEdition
--	ConferenceEdition	Constant	JournalIssue	Constant	Anàleg a EditedBook i ConferenceEdition
--	JournalVolume	Permanent	JournalIssue	Constant	Anàleg a ConferenceSeries i ConferenceEdition
--	Journal	Permanent	JournalVolume	Constant	Anàleg a ConferenceSeries i ConferenceEdition
--	JournalIssue	Constant	JournalPaper	Constant	Anàleg a EditedBook i BookSection
--	JournalIssue	Constant	JournalSection	Constant	Anàleg a JournalVolume i JournalIssue
--	JournalSection	Constant	JournalPaper	Constant	Anàleg a BookSection i BookChapter
--	BookSeries	Permanent	BookSeriesIssue	Constant	És un cas equivalent a ConferenceEdition i ConferenceSeries.
--	BookSection	Constant	BookSeriesIssue	Constant	Anàleg a BookSection i BookChapter
--	BookChapter	Constant	BookSection	Constant	Anàleg a EditedBook i BookSection
--	BookSeriesIssue	Constant	BookChapter	Constant	Anàleg a EditedBook i BookChapter
IsAuthorOf	Person	Permanent	AuthoredPublication	Constant	Idem que isEditorOf, però amb AuthoredPublication
/Publishes	Person	Permanent	Publication	Constant	El tipus de relació, és derivada per unió de IsAuthorOf i IsEditorOf

Atributs

Per a tots els atributs, el motiu de ser permanents o constants és molt evident. Per això no s'ha indicat la columna de motiu, tal i com s'ha fet en la resta de casos, atès que no aportaria informació útil.

<i>Nom</i>	<i>Tipus d'entitat a la que pertany</i>	<i>Tipus</i>
Name	Person	Variable
HomePage	Person	Variable
/numPublicacions	Person	Variable
Title	Publication	Constant
/year	Publication	Constant
/edition	Publication	Constant
NumPages	Book	Constant
HomePage	Book	Variable
Publisher	Book	Constant
Publication	Book	Constant
ISBN	Book	Constant
IniPage	BookChapter	Constant
EndPage	BookChapter	Constant
/conferencePaper	BookChapter	Constant
IniPage	JournalPaper	Constant
EndPage	JournalPaper	Constant
/conferencePaper	JournalPaper	Constant
Title	JournalSection	Constant
Order	JournalSection	Constant
Number	JournalIssue	Constant
Year	JournalIssue	Constant
Month	JournalIssue	Constant

Nom	Tipus d'entitat a la que pertany	Tipus
/numPages	JournalIssue	Constant
Volume	JournalVolume	Constant
Title	Journal	Variable
ISSN	Journal	Constant
Scronym	ConferenceSeries	Variable
Name	ConferenceSeries	Variable
Title	ConferenceEdition	Constant
Year	ConferenceEdition	Constant
City	ConferenceEdition	Constant
Country	ConferenceEdition	Constant
HomePage	ConferenceEdition	Variable
Title	BookSection	Constant
Order	BookSection	Constant
Number	BookSeriesIssue	Constant
Id	BookSeries	Constant
Order	BookSeries	Constant

7.4 Article presentat i acceptat en JISBD07

Definición, importancia y especificación en UML de las restricciones de integridad *constante* y *permanente*

Raquel Pau, Antoni Olivé
Universidad Politécnica de Cataluña
Dep. Lenguajes y Sistemas Informáticos
e-mail: raquelpau@gmail.com, olive@lsi.upc.edu

Palabras clave

Modelización conceptual, restricciones de integridad temporales, UML.

Resumen

Este artículo se centra en el estudio de dos restricciones temporales, denominadas constante y permanente, que se aplican a tipos de entidad, a tipos de relación y a participaciones en tipos de relación. El artículo proporciona una definición formal de estas restricciones, utilizando la lógica temporal, y un análisis de algunas de sus propiedades. A continuación, y basándose en las conclusiones del estudio de tres casos reales, se expone que estas restricciones aparecen muy a menudo en los esquemas conceptuales, por lo que tienen una gran importancia práctica. Esta importancia justifica el uso de una construcción lingüística específica que facilite su definición. El artículo explica como pueden definirse estas dos restricciones en UML mediante el uso de estereotipos.

1. Introducción

De acuerdo con el principio del 100%, o de completitud [14], un esquema conceptual debe incluir la definición de todas las restricciones de integridad (para abreviar, restricciones) que deben satisfacer la base de información y los eventos de un sistema de información. Por este motivo, todos los lenguajes completos de modelización conceptual incluyen un (sub)lenguaje para definir restricciones.

En el caso general, una restricción puede ser cualquier expresión válida del lenguaje correspondiente. No obstante, hay algunas clases particulares de restricciones que se definen en muchos esquemas conceptuales y a menudo varias veces en un mismo esquema. Naturalmente, estas restricciones pueden definirse en el lenguaje general de definición de restricciones, pero resulta práctico utilizar construcciones lingüísticas especiales para ellas que faciliten su definición. Las clases más importantes son las restricciones de integridad referencial y las de

cardinalidad, que ya estaban identificadas en el artículo original del modelo entidad-relación [2] y que todos los lenguajes tratan de forma especial. El lenguaje que probablemente identifica más restricciones particulares es el ORM [6].

Las restricciones particulares más populares son estáticas. No obstante, hay algunas restricciones temporales que aparecen tan a menudo en los esquemas conceptuales que también justifican un tratamiento especial. En este artículo estudiamos dos de ellas, que denominamos *constante* y *permanente*, y que pueden aplicarse a tipos de entidad, a tipos de relación y a participaciones en tipos de relación.

Estas restricciones no son nuevas. Se mencionan con alguna variación en diversos textos (como, por ejemplo, en [4]) y son recogidas en [7], pero no existe una definición completa y formal. El primer objetivo de este artículo (Sección 2) es definir formalmente estas restricciones, utilizando la lógica temporal, y analizar sus propiedades. El segundo objetivo es determinar la importancia de estas restricciones en la práctica para comprobar si justifican o no un tratamiento especial. En la Sección 3 resumimos los trabajos realizados y los resultados obtenidos. El tercer y último objetivo (Sección 4) es argumentar que las restricciones propuestas no pueden definirse con los elementos proporcionados por UML, requiriéndose una extensión basada en el uso de estereotipos, en la línea indicada en [7]. En el artículo utilizaremos ejemplos basados en el caso de estudio de eBay [10].

2. Definición

En esta sección introducimos (2.1) los conceptos y la notación básicos que usaremos, definimos (2.2 y 2.3) formalmente las restricciones estudiadas y analizamos (2.4 y 2.5) algunas de sus propiedades.

2.1. Conceptos y notación básica

Usaremos la lógica de primer orden, y en particular la lógica temporal, para las definiciones formales [3]. En lógica temporal, se considera que la base de información pasa por una serie de estados $\langle \sigma_0, \dots, \sigma_n \rangle$.

Un estado de la base de información contiene los hechos de entidades y relaciones ciertas en ese estado. Se puede considerar que hay un estado para cada instante del periodo de vida del sistema, pero no es obligatorio. De hecho, es más frecuente suponer que cada vez que cambian uno o más hechos, hay un cambio de estado de σ_i a σ_{i+1} . [8]

La representación lógica de estos estados no incluye el argumento temporal, porque éste es implícito. Esto conlleva al uso de operadores con semántica temporal. A continuación resumimos el significado de los operadores universales empleados en las fórmulas de este trabajo [1, 3]:

- $\Box A$ (siempre en el futuro)

La fórmula $\Box A$ es cierta en el estado σ_i si, y sólo si, para todos los estados σ_j , $i < j < n$ A es cierta en el estado σ_j .

- $\blacksquare A$ (siempre en el pasado)

La fórmula $\blacksquare A$ es cierta en el estado σ_i si, y sólo si, para todos los estados σ_j , $0 < j < i$ A es cierta en el estado σ_j .

- $\bullet A$

La fórmula A es cierta en el estado σ_i si y sólo si $i > 0$ y A es cierta en el estado σ_{i-1} .

- $A \text{ until } B$

La fórmula $A \text{ until } B$ es cierta en el estado σ_i si, y sólo si, para algún estado σ_j , $i \leq j \leq n$, B es cierta en el estado σ_j y, para todo k , $i < k < j$, A es cierta en σ_k .

- $A \text{ since } B$

La fórmula $A \text{ since } B$ es cierta en el estado σ_i si, y sólo si, para algún estado σ_j , $0 \leq j \leq i$, B es cierta en el estado σ_j y, para todo k , $j < k < i$ A es cierta en el estado σ_k .

Designaremos por $E(e)$ el hecho que e es una instancia del tipo de entidad E en un cierto estado. Por ejemplo, si *Artículo* es un tipo de entidad, *Artículo (micoche)* significa que la entidad *micoche* es una instancia de *Artículo*.

Representaremos mediante $R(p_1:E_1, \dots, p_n:E_n)$: el esquema de un tipo de relación llamado R con los tipos de entidad participantes E_1, \dots, E_n jugando los roles p_1, \dots, p_n respectivamente. Los atributos son considerados tipos de entidad binaria. Si se omite un role p_i se supone que es igual al tipo de entidad participante E_i empezando en minúscula. Designaremos por $R(e_1, \dots, e_n)$ el hecho que las entidades e_1, \dots, e_n participan en una instancia de R en un cierto estado. Por ejemplo, si *Posee(propietario:Vendedor,*

Artículo) es un tipo de relación, entonces *Posee(juan, micoche)* significa que el vendedor *juan* es el propietario del artículo *micoche*.

2.2. Tipos de entidades constante y permanente

Un tipo de entidad E es *permanente* si sus instancias nunca pueden dejar de ser instancia de E . La población de un tipo de entidad *permanente* nunca puede decrecer. Expresado en lógica temporal (se supone que todas las variables que no están cuantificadas están cuantificadas universalmente en la cabecera de la fórmula):

$$E(e) \rightarrow \Box E(e) \quad (1)$$

Un ejemplo de tipo de entidad *permanente* es *ArtículoOfrecido*. Las instancias de este tipo son los artículos que se han ofrecido en alguna ocasión. A partir del momento que un artículo es ofrecido, lo será siempre más.

Un tipo de entidad E es *constante* si su población es siempre la misma. Expresado en lógica temporal:

$$E(e) \rightarrow \blacksquare E(e) \wedge \Box E(e) \quad (2)$$

Es fácil ver, que los tipos de entidad *constantes*, también son *permanentes*.

Los tipos de datos (*DataTypes*, que en UML incluyen los *enumeration*) son *constantes* por definición.

Por ejemplo, en eBay un *extra* es un servicio que ofrece la empresa para una venta más rápida de un artículo (destacado, con fotografías,...). Los extras son instancia del tipo de entidad *Extra* que es *constante*.

2.3. Participación y tipos de relación constante y permanente

Un tipo de relación $R(p_1:E_1, \dots, p_n:E_n)$ es *permanente* con respecto a un participante p_i , siendo $1 \leq i \leq n$, si todas las instancias de R en las que una instancia e_i de E_i participa con el rol p_i nunca dejan de existir durante el intervalo temporal en el que e_i es instancia de E_i . Expresado en lógica temporal:

$$R(e_1, \dots, e_n) \rightarrow (R(e_1, \dots, e_n) \text{ until } \neg E_i(e_i)) \quad (3)$$

Un ejemplo. En eBay, siempre se tiene constancia de todos los comentarios que hace un usuario sobre otros. Así se crea una cierta imagen de confianza o *feedback* de los usuarios para la compra-venta de artículos. Luego, el tipo de relación *HechoPor*

(Comentario, emisor: Usuario), es permanente con respecto a la participación del rol emisor.

Un tipo de relación $R(p_1:E_1, \dots, p_n:E_n)$ es *constante* con respecto al participante p_i , siendo $1 \leq i \leq n$, si todas las instancias de R en las que una instancia e_i de E_i participa con el rol p_i son las mismas durante el intervalo temporal en el que e_i existe. Expresado en lógica temporal:

$$R(e_1, \dots, e_n) \rightarrow (R(e_1, \dots, e_n) \text{ since } (\bullet E_i(e_i))) \wedge (R(e_1, \dots, e_n) \text{ until } \neg E_i(e_i)) \quad (4)$$

En eBay, los comentarios los hace una sola persona y no pueden cambiar de autor. Luego, el mismo tipo de relación anterior *HechoPor* es *constante* con respecto a la participación del rol *comentario*.

Es importante remarcar que, si $R(p_1:E_1, \dots, p_n:E_n)$ es *permanente* respecto un participante, ésta no tiene porqué ser permanente con respecto el resto de participantes.

Por ejemplo, en eBay siempre se guardan todas las búsquedas. Por consiguiente, *Búsqueda* es un tipo de entidad permanente. Las búsquedas se pueden hacer dentro de una categoría, pero no es obligatorio. El administrador del sistema, gestiona las categorías, y puede eliminar una cierta categoría siempre y cuando no contenga artículos para vender.

El tipo de relación: *BuscaEn* (*Búsqueda*, *Categoría*), es permanente respecto *categoría* pero no respecto *búsqueda*. Si el administrador elimina una instancia de tipo *Categoría* se han de eliminar el conjunto de relaciones del tipo *BuscaEn* en las que participa aquella categoría, pero no las instancias de *Búsqueda*.

Los tipos de relaciones derivadas que son constantes respecto un participante, tienen una peculiaridad, y es que, su población vendrá determinada por el valor de la regla de derivación del estado en el que se crea una instancia del tipo de dicho participante. Esto se aplica también para los atributos derivados *constantes*.

Un tipo de relación $R(p_1:E_1, \dots, p_n:E_n)$ es *permanente* si lo es con respecto a todos sus participantes. Expresado en lógica temporal:

$$R(e_1, \dots, e_n) \rightarrow R(e_1, \dots, e_n) \text{ until } \neg E_1(e_1) \quad \dots \quad R(e_1, \dots, e_n) \rightarrow R(e_1, \dots, e_n) \text{ until } \neg E_n(e_n) \quad (5)$$

Por ejemplo, el tipo de relación *HechoPor* (*Comentario*, *emisor:Usuario*), es permanente con respecto a la participación del rol *emisor* y constante respecto del rol *comentario*. Dado que la restricción

constante es más restrictiva que permanente, éste es un tipo de relación permanente.

Un tipo de relación $R(p_1:E_1, \dots, p_n:E_n)$ es *constante* si lo es con respecto a todos sus participantes. Expresado en lógica temporal:

$$R(e_1, \dots, e_n) \rightarrow (R(e_1, \dots, e_n) \text{ since } (\bullet E_1(e_1))) \wedge (R(e_1, \dots, e_n) \text{ until } \neg E_1(e_1)) \quad \dots \quad R(e_1, \dots, e_n) \rightarrow (R(e_1, \dots, e_n) \text{ since } (\bullet E_n(e_n))) \wedge (R(e_1, \dots, e_n) \text{ until } \neg E_n(e_n)) \quad (6)$$

En eBay el tipo de entidad *IncrementoDePuja* representa los datos para saber el incremento que, dado un intervalo de precios, ha de tener la siguiente puja para que pueda ser ganadora de una subasta. Las instancias de *IncrementoDePuja* y el valor de los sus atributos no cambian. El atributo *Inicio* representa el tipo de relación *Inicio(IncrementoDePuja, Importe)*, donde *Importe* es un *DataType*. En este caso, las dos participaciones son constantes de lo que se deduce que el tipo de relación también lo es.

2.4. Propiedades

En este apartado analizamos la interacción de las restricciones constante y permanente de tipos de relación y de los tipos de entidad que participan en ellas. Por falta de espacio, omitimos las demostraciones formales, que pueden encontrarse en [9]. Para simplificar la exposición consideramos únicamente tipos de relación binarias, pero la extensión al caso general es simple.

Propiedad 1: Si E_1 es un tipo de entidad permanente y $R(p_1:E_1, p_2:E_2)$ es un tipo de relación permanente respecto al participante p_1 , entonces R es permanente respecto al participante p_2 .

En eBay, *Usuario* es un tipo de entidad permanente y su participación en *HechoPor* (*Comentario*, *emisor:Usuario*) es permanente bajo el rol de *emisor*. De acuerdo con esta propiedad, *HechoPor* debe ser permanente con respecto a *comentario*.

La justificación es la siguiente: Cuando la participación p_1 es permanente sólo se pueden eliminar instancias de R si se elimina la instancia de E_1 correspondiente. En este caso, E_1 es permanente. Por lo tanto, no pueden eliminarse sus instancias. Esto conlleva a que p_2 sea permanente y, por extensión, que R sea permanente.

Propiedad 2: Si E_1 es un tipo de entidad constante y $R(p_1:E_1, p_2:E_2)$ es un tipo de relación constante

respecto al participante p_1 , el tipo de relación es constante respecto al participante p_2 .

En eBay, *IncrementoDePuja* es un tipo de entidad constante y su participación en *Inicio(IncrementoDePuja, Importe)* es constante bajo el rol de *incrementoDePuja*. De acuerdo con esta propiedad, *Inicio* debe ser constante con respecto a *Importe*.

La justificación es la siguiente: Cuando la participación p_1 es constante sólo se pueden añadir o eliminar instancias de R al crear o eliminar la instancia de E_1 correspondiente. En este caso, E_1 es constante. Por lo tanto, no se pueden ni crear ni eliminar sus instancias. Esto conlleva a que p_2 sea constante y, por extensión, que R sea constante.

Propiedad 3: Si E_1 es un tipo de entidad permanente y $R(p_1:E_1, p_2:E_2)$ es un tipo de relación donde:

- La participación p_1 es permanente.
 - La participación de E_2 en el rol p_2 de R es obligatoria.
- Entonces, E_2 también es un tipo de entidad permanente.

En eBay, el tipo de entidad *Usuario* es permanente y su participación en *HechoPor (Comentario, emisor:Usuario)* en el rol *emisor* es permanente. Por otra parte, todas las instancias de *Comentario* deben estar hechas por un usuario, por lo que la participación de *Comentario* en *HechoPor* es obligatoria. De acuerdo con esta propiedad, *Comentario* debe ser permanente.

De acuerdo con la propiedad 1, R es permanente respecto de p_2 y, por consiguiente, R es permanente. Si se eliminase una instancia cualquiera de E_2 , por la restricción de integridad referencial también deberían eliminarse las una o más instancias de R en que participa. Como que tal eliminación no es posible, las instancias de E_2 no pueden eliminarse, por lo que E_2 es permanente.

Propiedad 4: Si E_1 es un tipo de entidad constante y $R(p_1:E_1, p_2:E_2)$ es un tipo de relación donde:

- La participación p_1 es constante.
- La participación de E_2 en el rol p_2 de R es obligatoria.

Entonces, E_2 también es un tipo de entidad constante.

En eBay, el tipo de entidad *Importe* es constante y su participación en *Inicio (IncrementoDePuja, Importe)* en el rol *importe* es constante. Por otra parte, todas las instancias de *IncrementoDePuja* indican un importe de inicio sobre el que aplicar el incremento. De acuerdo con esta propiedad, *IncrementoDePuja* debe ser constante.

De acuerdo con la propiedad 1, R es constante respecto de p_2 y, por consiguiente, R es constante. Si se crease una instancia cualquiera de E_2 , dado que su participación es obligatoria, también se debería crear una o más instancias de R . Tal creación no es posible: E_1 es constante y su participación también. Luego, E_2 ha de ser constante.

2.5. Consistencia con el esquema de comportamiento

En este punto se presentan las condiciones que imponen las restricciones constante y permanente en el esquema de comportamiento para que sea consistente con el esquema estructural. Son las siguientes:

1. No puede existir un evento de dominio que cree instancias de un tipo de entidad que sea constante.
2. Sea el tipo de relación $R(p_1:E_1, \dots, p_n:E_n)$. Si E_i tiene una participación constante en R , entonces sólo se pueden crear relaciones de R cuando se crea una instancia de E_i .
3. No puede existir un evento de dominio que elimine instancias de un tipo de entidad permanente (y consecuentemente, constante).
4. Sea el tipo de relación $R(p_1:E_1, \dots, p_n:E_n)$. Si E_i tiene una participación permanente en R , entonces sólo se pueden eliminar las relaciones de R en que participa una entidad e_i cuando se elimina dicha entidad.
5. Sea el tipo de relación $R(p_1:E_1, \dots, p_n:E_n)$ y E_j un tipo de entidad que tiene algún rol con participación permanente en R . Si eliminamos una entidad $e_i \in E_i$, tal que existe alguna relación de R en que participa e_i , también eliminaremos todas las instancias de E_j que participaban en alguna relación con e_i .
6. No puede haber un evento de dominio que modifique el valor de un atributo de una entidad ya existente si es constante.

3. Importancia práctica

3.1. Motivación

Un lenguaje de modelización conceptual permite representar gráficamente tipos de entidad y de relación mediante unos ciertos símbolos. Disponer de una simbología específica para expresar restricciones de integridad aporta, en caso de que éstas sean muy

frecuentes, una mayor agilidad en el desarrollo de esquemas conceptuales. Un claro ejemplo de esta realidad sucede con las restricciones de cardinalidad.

Por consiguiente, se ha considerado interesante investigar el grado de aparición de las restricciones constante y permanente en el esquema estructural de tres casos de estudio. Posiblemente, estos casos de estudio sean representativos de otros sistemas de información del mundo de la industria.

Para realizar esta tarea se ha obtenido la especificación de cada caso de estudio. A continuación, para cada elemento del esquema estructural se ha determinado y justificado en [9] si cumplía las restricciones de constante o permanente teniendo en cuenta las anteriores propiedades. Finalmente, a partir de los resultados se ha calculado la frecuencia de estas restricciones.

Por tanto, a partir de las cifras calculadas podremos concluir si sería conveniente extender un lenguaje de modelización conceptual para poder representarlas de forma específica.

3.2. Casos de estudio

Los casos de estudio escogidos son:

- *eBay*: empresa de mercado electrónico donde se realizan subastas. La gente puede comprar y vender mercancías y servicios, por todo el mundo, a través suyo. [10]
- *EU-Rent Car Rentals*: empresa ficticia (pero popular) que se dedica a alquilar coches. Tiene sucursales en distintos países. Sus servicios incluyen, entre otros, la reserva anticipada de coches y distintas formas de pago [5].
- *DBLP*: sistema web que originalmente contenía información bibliográfica de sistemas con bases de datos y de programación lógica. Actualmente DBLP se está extendiendo, proporcionando información bibliográfica de gran parte de revistas y actas de ciencias de computación.

3.3. Resultados

La frecuencia de apariciones de las restricciones constante y permanente para cada caso de estudio, se desglosa en los siguientes grupos de elementos:

- Tipos de entidad (T. de entidad excluyendo DataTypes).

- Tipos de relación (T. de relación).
- Participaciones (Participaciones).
- Atributos (Atributos).

Dando valor a las siguientes variables de salida:

Total (Total): Cuyo valor corresponde al total de elementos de cada grupo.

Permanente (Perm): Cuyo valor corresponde al número de elementos de cada grupo que cumplen la restricción de permanente.

Constante (Const): Cuyo valor corresponde al número de elementos de cada grupo que cumplen la restricción de constante. Un atributo, se ha considerado constante, si lo es la participación del tipo de entidad que contiene el atributo.

Porcentaje (%): Cuyo valor corresponde al porcentaje de elementos de cada conjunto que cumplen las restricciones de constante o permanente.

<i>EU-Rent Car Rentals</i>	Total	Perm.	Const.	%
T. de entidad	56	31	3	61
T. de relación	59	25	0	42
Participaciones	119	29	18	39
Atributos	69	0	31	44

Tabla 1. Resultados del caso de estudio EU-Rent Car Rentals.

<i>eBay</i>	Total	Perm.	Const.	%
T. de entidad	54	42	2	81
T. de relación	34	15	0	44
Participaciones	77	36	5	53
Atributos	109	0	58	53

Tabla 2. Resultados del caso de estudio eBay.

<i>DBLP</i>	Total	Perm.	Const.	%
T. de entidad	17	15	0	88
T. de relación	18	7	11	100
Participaciones	36	7	29	100
Atributos	38	0	30	78

Tabla 3. Resultados del caso de estudio DBLP

3.4. Conclusiones

El porcentaje de aparición de las restricciones constante y permanente es muy elevado en los dos casos de estudio. Sobrepasa el 30% en todos los casos. Cabe el riesgo que se haya dado la particularidad que estos dos casos de estudio sean de los pocos en los que consigan estas cifras. De todas formas, insistimos que se han cogido ejemplos reales, y posiblemente, estos tengan muchas similitudes con otros sistemas del mundo de la industria. Por consiguiente, el riesgo es menor.

Ya se explicó que si un tipo de restricción aparece habitualmente, posiblemente sea conveniente poderla representar gráficamente, en un lenguaje de modelización conceptual. Así logramos agilizar la tarea de desarrollo del analista. Luego, vistos los porcentajes detectados, en este trabajo se propone una representación en UML para este tipo de restricciones.

4. Especificación en UML

UML es un lenguaje que puede usarse, y se usa, para la modelización conceptual, en el que su metaesquema no ofrece ni en la versión actual 2.1 ni en anteriores como la 1.4 una manera de expresar las restricciones de integridad constante y permanente. UML ofrece ciertos valores para la propiedad de cambiabilidad, que está ligada a las restricciones de integridad constante y permanente.

4.1. La cambiabilidad en la versión 2.1 de UML

La versión 2.1 de UML ha modificado substancialmente la definición de cambiabilidad proporcionada en la versión 1.4. Ésta tiene forma de un atributo booleano llamado *isReadOnly* y sólo se puede aplicar a *properties*.

En el metaesquema de la versión 2.1 de UML, el tipo de entidad *Property* tiene como instancias las participaciones de asociaciones y atributos.

El valor cierto en el atributo *isReadOnly* para el elemento marcado, indica que éste se podrá leer y no escribir [13]. Tiene dos restricciones de integridad relacionadas.

- La propiedad *isNavigable* puede ser marcada como *isReadOnly*. Su expresión en OCL :

context Property inv:

isReadOnly implies isNavigable

- Si una *property* tiene *isDerivedUnion* a cierto, entonces *isReadOnly* es cierto. Su expresión en OCL:

Tèsi de Màster

Restriccions d'integritat temporals constant i permanent

Raquel Pau

Universitat Politècnica de Catalunya

context Property inv:

isDerivedUnion implies isReadOnly

En [11] ya se indica que la cambiabilidad es un concepto definido de forma imprecisa. Una *property* ha de ser modificable mientras es inicializada, pero el concepto de inicialización no está definido en UML. También se cita en [11] que un rango de valores, como en UML1, pero con más opciones, sería mejor que un simple valor booleano.

Podríamos llegar a pensar que es equivalente representar mediante *isReadOnly* las restricciones de integridad constante de un participante respecto un tipo de relación, y consecuentemente los tipos de relación constante. No es del todo cierto. Por una parte, que un atributo sólo se pueda leer no implica que no pueda cambiar, ya que éste puede ser derivado. Además, si un atributo es derivado por unión, no implica que éste sea constante, tal y como afirma la segunda restricción. Luego, no podemos representar las restricciones constantes mediante *readOnly*.

4.2. Propuesta ofrecida

En UML hay un mecanismo llamado *stereotype* para definir nuevas restricciones predefinidas. Un *stereotype* es una clase cuyas instancias extienden las características definidas en un elemento del modelo. Estas nuevas características son información adicional que puede ser usada para distintos propósitos. Entonces, para una correcta representación de las restricciones de integridad constante y permanente proponemos incluirlas como estereotipos del elemento *Constraint*.

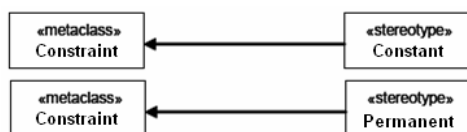


Figura 1. Estereotipos *constant* y *permanent*

En UML para representar que una restricción C la ha de cumplir una cierta población, se hace mediante {C}. Por consiguiente, la representación de un tipo de entidad *constante* o *permanente* en UML, se conseguiría escribiendo {«constant»} o {«permanent»}. Esta notación tendría que aparecer dentro del recuadro del tipo de entidad que ha de cumplir dicha restricción.

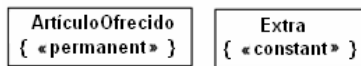


Figura 2. Ejemplo de tipos de entidad en UML

Análogamente, la representación de una participación constante o permanente respecto un tipo de relación se indicaría escribiendo {«constant»} o {«permanent»} en el extremo de la relación donde aparece dicho participante. Para los atributos se emplearía la misma simbología al lado derecho donde aparece el tipo del atributo.



Figura 3. Ejemplo de tipos de relación en UML

5. Conclusiones

El objetivo de este artículo se ciñe a dos restricciones temporales, denominadas *constante* y *permanente*. Estas restricciones se aplican a tipos de entidad, a tipos de relación y a participaciones en tipos de relación. En el artículo hemos proporcionado una definición formal y completa de ambas restricciones, usando la lógica temporal. Por otra parte, hemos presentado un conjunto de propiedades de estas restricciones y su impacto en el esquema del comportamiento. La definición formal y de sus propiedades ayuda a validar esquemas estructurales y de comportamiento.

Mediante el análisis de los esquemas conceptuales de tres casos de estudio reales, hemos llegado a la conclusión que estas restricciones aparecen con mucha frecuencia en la práctica. Esta conclusión justifica que los lenguajes de modelización conceptual proporcionen elementos específicos para su definición. Hemos visto que el UML no los proporciona, por lo que está justificado ampliar este lenguaje mediante nuevos estereotipos, que también hemos descrito.

El trabajo aquí descrito se puede continuar en varias direcciones. En primer lugar, puede ser interesante proporcionar una definición alternativa, usando una representación de los hechos de entidad y de relación con tiempo explícito, en lugar del tiempo implícito de la lógica temporal. En segundo lugar, merece la pena analizar otros esquemas conceptuales de sistemas reales para corroborar o no que las restricciones estudiadas merecen un tratamiento particular. En tercer lugar, pensamos implementar estas

restricciones en un entorno de modelización concreto, incluyendo el control de las propiedades encontradas. Finalmente, pensamos que es interesante desarrollar algún patrón de diseño para el control de estas restricciones.

Agradecimientos

La realización de este trabajo ha conestado con la ayuda del Ministerio de Educación y Ciencia, proyecto TIN2005-06053

Referencias

- [1] Bellini, P., Mattolini, R., Nesi, P. Temporal Logics for Real-Time System Specification. ACM Computing Surveys. 2000.
- [2] Chen PPS The Entity-Relationship model: Towards a unified view of data. ACM Trans. Database Syst. 1(1):9-36, 1976
- [3] Chomicki, J. Saake, G.. Logics for databases and Information systems. Kluwer Academia Publishers. 1998
- [4] Costal, D., Olivé, A., Sancho, M-R.: Temporal Features of Class Populations and Attributes in Conceptual Models. ER 1997: 57-70
- [5] Frías, L., Queralt, A., Olivé, A. EU-Rent Car Rentals Specification. LSI-03-59-R. 2003.
- [6] Halpin T Information modeling and relational databases. From conceptual análisis to logical design. Morgan Kaufmann Publishers. 2001.
- [7] Olivé, A., Conceptual Modeling of Information Systems. Springer Verlag. 2007.
- [8] Olivé, A. Modelització conceptual de sistemes d'informació. Edicions UPC. 2002.
- [9] Pau, R. Master Thesis. Les restriccions d'integritat temporals constant i permanent. 2007.
- [10] Queralt, A. Reenginyeria d'un sistema de mercat electrònic. 2003. UPC
- [11] Rumbaugh, J., Jacobson, I., Booch, G. The Unified Modeling Language Reference Manual. Second Edition. Addison- Wesley. 2004.
- [12] Unified Modeling Language Specification. Version 1.4.2. formal/04-07-02.
- [13] Unified Modeling Language: Superstructure. Version 2.1.1. Formal/2007-02-03.
- [14] Van Griethuysen JJ. Concepts and terminology for the conceptual schema and the information base. ISO TC97/SC5/WG3. 1982.